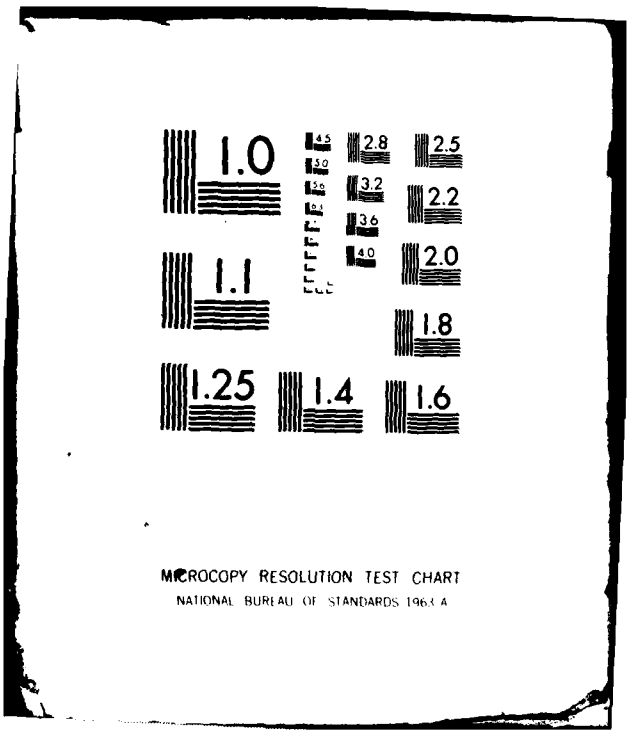
AD-A088 751 CALIFORNIA UNIV BERKELEY ELECTRONICS RESEARCH LAB F/G 12/1
ACCURATE COMPUTATION OF DIVIDED DIFFERENCES.(U)
MAY 80 A C MCCURDY N00014-76-C-0013
UNCLASSIFIED UCB/ERL-M80/28 NL

1 OF 2
AD A088751


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A088 757	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Accurate Computation of Divided Differences		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER Memo. No. UCB/ERL M80/28
7. AUTHOR(s) A. C. McCurdy		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0013
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California Electronics Research Laboratory College of Engineering, Berkeley, CA 94722		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N, RR 01402, RR0140201, NR 044-324
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Mathematics Program, Code 432 800 N. Quincy St., Arlington, VA 22217		12. REPORT DATE 28 May 1980
		13. NUMBER OF PAGES 155
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Numerical Analysis
Divided Differences
Recurrence Schemes
Computational Methods

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
The standard reccurence scheme does not always yield accurate divided differences in finite precision arithmetic. When the function of interest is known analytically and/or its values are easily calculated, methods other than the recurrence scheme can be used. In particular, a table of divided differences can be regarded as a function of a special bidiagonal matrix. Formulas and computational techniques suitable for computing matrix functions may, thus, be exploited for divided differences.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Divided difference tables of the exponential function are profitably treated as the exponential of a special matrix. This approach is good precisely when the standard recurrence is bad, namely when the abscissae of the divided differences are close. When the abscissae are scaled down by powers of 2, the result scaled divided difference table may be squared to give the wanted table. For real abscisae this scaling and squaring technique, in combination with the standard recurrence where suitable, yields a hybrid algorithm which permits computation of any exponential divided difference to an accuracy dependent only on the order of the difference. For appropriate arrangements of complex abscissae, such as conjugate pairs, a similar result is established. A good way to compute the exponential of a real square matrix A is to use the Newton divided difference interpolating polynomial. Our algorithm finds an important application in computing accurately the coefficients of this polynomial.

(12)

(4) ACCURATE COMPUTATION OF DIVIDED DIFFERENCES.

by

(10) Allan Charles/McCurdy

(12) 1621

(9) Doctoral thesis,

(15) N00014-76-C-0013

(14) Memorandum No. UCB/ERL-M80/28

(16) KR0217+03

(11) 28 May 1980

(17) KR021402.14

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Accurate Computation of Divided Differences

By

Allan Charles McCurdy
B.S. (Stanford University) 1968
M.S. (New York University) 1970
C.Phil. (University of California) 1978

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Applied Mathematics

in the

GRADUATE DIVISION

OF THE

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved:

B. N. Parlett April 17, 1980
Chairman Date

M. H. Protter

.....

.....

Accurate Computation of Divided Differences

by

Allan Charles McCurdy

ACCURATE COMPUTATION OF DIVIDED DIFFERENCES

Allan Charles McCurdy

Abstract

The standard recurrence scheme does not always yield accurate divided differences in finite precision arithmetic. When the function of interest is known analytically and/or its values are easily calculated, methods other than the recurrence scheme can be used. In particular, a table of divided differences can be regarded as a function of a special bidiagonal matrix. Formulas and computational techniques suitable for computing matrix functions may, thus, be exploited for divided differences.

Divided difference tables of the exponential function are profitably treated as the exponential of a special matrix. This approach is good precisely when the standard recurrence is bad, namely when the abscissae of the divided differences are close. When the abscissae are scaled down by powers of 2, the resulting scaled divided difference table may be squared to give the wanted table. For real abscissae this scaling and squaring technique, in combination with the standard recurrence where suitable, yields a hybrid algorithm which permits computation of any exponential divided difference to an accuracy dependent only on the order of the difference. For appropriate arrangements of complex abscissae, such as conjugate pairs, a similar result is established. A good way to compute the exponential of a real square matrix A is to use the Newton divided difference interpolating polynomial. Our algorithm finds an important application in computing accurately the coefficients of this polynomial.

Acknowledgements

I wish to express my appreciation to Prof. Beresford Parlett for giving his time, help and support during the writing of this thesis. Also, I wish to thank Prof. M. Protter and Prof. E. Polak for their comments. This research was sponsored by the Office of Naval Research contract N00014-76-C-0013 to the Electronics Research Laboratory, University of California, Berkeley.

Contents

ACCURATE COMPUTATION OF DIVIDED DIFFERENCES

A function of a matrix, $f(A)$, may be defined in terms of a polynomial which interpolates f at A's eigenvalues. One such interpolating polynomial is derivable from Newton's divided difference interpolation formula. Coefficients in this interpolating polynomial are divided differences of f at the eigenvalues of A. Thus $f(A)$ may be represented in terms of divided differences of f.

The opposite is also true, though this is not widely known. That is, divided differences of f may be represented in terms of the function of a special matrix. Matrix functions and divided differences, then, are profitably studied together. In particular, techniques used to compute matrix functions may be exploited to study and calculate divided differences. The exploitation of matrix function theory for the study of divided differences is the prime purpose here. In a number of cases it will lead to new methods for accurate computation of divided differences.

The first chapter is a brief introduction to matrix functions. The interpolating polynomial definition leads immediately to several matrix theoretic properties of $f(A)$, for example A and $f(A)$ commute. The Newton divided difference polynomial explicitly shows the use of divided differences in defining $f(A)$. An extension to a divided difference series representation of $f(A)$ is given for holomorphic f.

The second chapter is a general study of divided differences. §2.1 introduces a new compact divided difference notation and lists, in this new notation, a number of facts about divided differences. For completeness, the following sections outline the classical approach to the study of divided differences and the advantages of an entirely different view of them as functions of their data points. §2.6 establishes the matrix function formula for divided difference tables. The remaining sections exploit this formula to develop series expansions for divided differences.

Chapter 3 is a study, in detail, of divided differences of the exponential function and methods for computing them. The special nature of $f = \exp$ gives its divided differences properties not shared by those of other functions. These properties are presented in §3.1. §3.2 develops bounds on exponential divided differences with real data. These bounds show how

errors grow in computing divided differences by the standard method. The following sections present, with error analyses, a Taylor series algorithm and a scaling and squaring algorithm for computing exponential divided differences. The latter is a direct consequence of representing the divided difference table as an exponential of a matrix. §3.5 then outlines a hybrid of those two algorithms and shows how real exponential divided differences can be computed with a bounded relative error. Of prime importance is the fact that the error bounds depend only on the order of the difference, not the data. Finally, the remaining sections study complex exponential divided differences, with particular attention paid to methods for computing divided differences with data consisting of conjugate pairs.

1. Matrix Functions

1.1 Definitions and representations of matrix functions.

The extension to matrices of the concept of function has led to several definitions of matrix function. Nonetheless, Rinehart [1955] has shown that all common definitions of matrix function are equivalent for functions holomorphic on a region containing the eigenvalues of the matrix. Since here we concentrate on holomorphic functions, we are free to choose a definition which makes presentation easiest. A definition of a function of a matrix in terms of interpolating polynomials has a natural relationship with divided differences. We choose this as our primary definition.

Let A be an $(n+1)\times(n+1)$ constant matrix whose elements may be complex numbers. We display the eigenvalues of the matrix A in the sequence $\Lambda_A \equiv \{\lambda_0,\ldots,\lambda_0,\lambda_1,\ldots,\lambda_1,\ldots,\lambda_l,\ldots,\lambda_l\}$ in which $l+1$ of the eigenvalues are distinct and each distinct eigenvalue occurs n_i+1 times, $i=0,1,\ldots,l$. Λ_A has $\sum_{i=0}^{l}(n_i+1)=n+1$ entries. The elements of Λ_A are just the roots of A's characteristic polynomial

$$\chi_A(\lambda)=(\lambda-\lambda_0)^{n_0+1}(\lambda-\lambda_1)^{n_1+1}\cdots(\lambda-\lambda_l)^{n_l+1}. \tag{1.1.1}$$

The definition we give for $f(A)$ requires simply that $f(\lambda)$ be defined for each $\lambda\in\Lambda_A$ when the eigenvalues are all distinct. To allow for multiple eigenvalues, however, we require that f be defined on Λ_A as follows.

Definition: The function f is said to be "defined on the characteristic values of A" when $f(\lambda_i)$, $f'(\lambda_i),\ldots,f^{(n_i)}(\lambda_i)$ are defined for each $i=0,1,\ldots,l$. For brevity, we denote this sequence of values by $f(\Lambda_A)$.

For any f satisfying this definition, $f(A)$ is defined in terms of an interpolating polynomial for f.

Function of a matrix. When f is defined on the characteristic values of A and p is any polynomial such that

$$p(\Lambda_A) = f(\Lambda_A),$$

then

$$f(A) \equiv p(A). \tag{1.1.2}$$

The polynomial p is an osculating interpolation polynomial for f on Λ_A. That is, $p(\lambda_i) = f(\lambda_i)$, $p'(\lambda_i) = f'(\lambda_i)$,..., $p^{(n_i)}(\lambda_i) = f^{(n_i)}(\lambda_i)$ for each $i = 0, 1, \ldots, l$. When the eigenvalues are distinct this definition of $f(A)$ becomes particularly simple, as then p is just an ordinary interpolating polynomial for f at the elements of Λ_A.

The rationale behind definition (1.1.2) is that for two functions f and g, $f(A)$ is indistinguishable from $g(A)$ when $f(\Lambda_A) = g(\Lambda_A)$. The sequence of zeros of $f(\lambda) - g(\lambda)$ includes Λ_A, the roots of $\chi_A(\lambda)$, and $\chi_A(A) = 0$ by the Cayley-Hamilton theorem. The interpolating polynomial p has degree at least n, since it must satisfy the $n+1$ conditions given in the definition.† An interpolating polynomial p may be chosen to satisfy additional conditions, but the degree of the polynomial is increased. We write p_n for the unique polynomial of least degree interpolating f on Λ_A.

p_n need not be the polynomial of least degree defining $f(A)$. The characteristic polynomial χ_A is an annihilating polynomial for A because $\chi_A(A) = 0$. However for some matrices A, there are polynomials of smaller degree which are also annihilating polynomials. The minimal polynomial μ_A is the non-trivial annihilating polynomial for A of least degree. If $\mu_A(\lambda)$ has degree $m+1$, $m \leqslant n$, it is possible to define $f(A)$ in terms of a m degree polynomial p_m which interpolates f at the $m+1$ roots of μ_A. Gantmacher [1959] uses this slightly more general approach in his definition of $f(A)$. The roots of $\mu_A(\lambda)$ are eigenvalues of A. For $m < n$ fewer derivatives of f need be specified, however μ_A and the multiplicities of its roots may be difficult and costly to obtain. Thus we shall not try to form $f(A) = p_m(A)$ for the smallest possible degree m. p_n can have significantly higher degree than p_m, see Fig. 1.2.1, but here we achieve greater simplicity in that less need be known about the matrix A.

†A polynomial of degree k can interpolate at, at most, $k+1$ points. In general $k+1$ points uniquely determine a polynomial of degree k; higher degree polynomials are not uniquely determined.

$$A = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 1 & 0 \\ -1 & 1 & 3 \end{bmatrix} \qquad \mu_A(\lambda) = (\lambda-1)(\lambda-2)$$

$$\chi_A(\lambda) = (\lambda-1)^2(\lambda-2)$$

Fig. 1.1.1: Degree of μ_A may be less than degree of χ_A.

The polynomial representation of $f(A)$ leads to several elementary, but very useful, consequences.

Similarity transformations. For any $(n+1) \times (n+1)$ nonsingular matrix P,

$$f(PAP^{-1}) = P \cdot f(A) \cdot P^{-1}. \tag{1.1.3}$$

In theory this permits performing all computations to form $f(A)$ on the simplest matrix similar to A, e.g. A's Jordan canonical form. In practice, however, the transformation matrix P may be difficult to compute accurately† or may be nearly singular. Some less simple form may be required. The triangular Schur form T, which is unitarily similar to A, eliminates the above objections.‡ However, $f(T)$ is not always simple to compute with accuracy.

Commutativity.

$$A \cdot f(A) = f(A) \cdot A \tag{1.1.4}$$

Parlett [1976] has presented a very fast method for computing functions of upper triangular matrices T based on this property. In brief, the diagonal of $f(T)$, which is also upper triangular, is computed directly;

$$f(T)_{i,i} = f(T_{i,i})$$

for each $i = 0, 1, \ldots, n$. Then successively by diagonals towards the upper right, the general recurrence is

†Kagström and Ruhe [1976] present an algorithm for computing the Jordan form, while Golub and Wilkinson [1976] discuss limitations on computing it accurately.
‡Wilkinson [1965] presents a detailed analysis of the QR algorithm which reduces A to T by a sequence of unitary similarity transformations; the algorithm is implemented in the *EISPACK* [Smith, 1974] collection of computer subroutines.

$$f(T)_{i,j} = \{\sum_{k=0}^{j-i-1} [f(T)_{i,i+k} \cdot T_{i+k,j} - T_{i,j-k} \cdot f(T)_{j-k,j}]\}/(T_{i,i} - T_{j,j}) \qquad (1.1.5)$$

where $i < j \leqslant n$. T may be A's Schur form; this recurrence may be used to form $f(A)$ by (1.1.3).

When f is symmetric in the real axis, that is $f(\bar{\zeta}) = \overline{f(\zeta)}$, polynomials interpolating f have real coefficients. We denote the conjugate transpose of A, $\bar{A}^{\mathrm{T}}$, by A^*.

Conjugate transpose. When f is symmetric in the real axis,

$$f(A^*) = f(A)^*. \qquad (1.1.6)$$

Expression (1.1.6) shows that conjugate symmetries in A are inherited by $f(A)$.

Formula (1.1.3) shows that $f(A)$, defined as in (1.1.2), may always be computed from A's Jordan canonical form. Conversely, we may wish to define $f(A)$ from the Jordan form by way of (1.1.3). This latter definition is more general than our polynomial definition, as the following shows.

The 2×2 identity matrix has, among others, the square roots

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The former root is representable by either definition; the latter is obtained by separately defining $\sqrt{1} = 1$ and $\sqrt{1} = -1$ on each Jordan block. The function is permitted to be multivalued, but only on separate Jordan blocks. The polynomial definition does not allow this, since polynomials are never multivalued.

Even the Jordan form definition of $f(A)$ is not the most general possible. For example a square root of

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{is} \quad \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

E. Cartan proposed a contour integral definition which applies to holomorphic functions f [Rinehart, 1955].

Cartan definition. If f is holomorphic inside and on a simple closed contour C enclosing Λ_A, then

$$f(A) \equiv \frac{1}{2\pi i}\int_C f(\zeta)\cdot(\zeta I - A)^{-1}\,d\zeta. \tag{1.1.7}$$

Additional representations of $f(A)$ are derivable from those just mentioned. Gantmacher [1959] and Rinehart [1955] discuss $f(A)$ in further detail. In the next section we present a particular polynomial representation for $f(A)$ and discuss related series representations.

1.2 The Newton polynomial of f(A), and series representations.

Because we are free to choose any polynomial interpolating f on Λ_A, definition (1.1.2) allows many representations of $f(A)$. There is, however, a unique interpolating polynomial p_n of least degree, though even this may be arranged in many ways.† One arrangement of p_n, which clearly illustrates the use of divided differences for defining matrix functions, is based upon Newton's divided difference formula for the interpolating polynomial, namely

$$p_n(\lambda) = \sum_{k=0}^{n} \blacktriangle_0^k f \cdot \prod_{j=0}^{k-1} (\lambda - \lambda_j) . \tag{1.2.1}$$

The coefficient $\blacktriangle_0^k f$ is the k-th order divided difference of f defined on the abscissae $\lambda_0, \lambda_1, \ldots, \lambda_k$. This compact divided difference notation is further explained in §2.1.

The first few terms of the interpolating polynomial (1.2.1), which we call a Newton polynomial, are

$$f(\lambda_0) + \blacktriangle_0^1 f \cdot (\lambda - \lambda_0) + \blacktriangle_0^2 f \cdot (\lambda - \lambda_0)(\lambda - \lambda_1) + \blacktriangle_0^3 f \cdot (\lambda - \lambda_0)(\lambda - \lambda_1)(\lambda - \lambda_2) + \cdots .$$

Because $p_n(\Lambda_A) = f(\Lambda_A)$ where $\Lambda_A = \{\lambda_0, \lambda_1, \ldots, \lambda_n\}$, the eigenvalues in Λ_A having been renumbered, $f(A)$ has the following representation.

Newton polynomial of f(A). When f is defined on the characteristic values of A,

$$f(A) = \sum_{k=0}^{n} \blacktriangle_0^k f \cdot \prod_{j=0}^{k-1} (A - \lambda_j I) . \tag{1.2.2}$$

Λ_A is the sequence of abscissae for the divided differences.

In §2.1 we shall see that the conditions on f necessary to define all the divided difference coefficients $\blacktriangle_0^k f$, $k = 0, 1, \ldots, n$, are exactly those required to assure the existence of some interpolating polynomial p_n. Thus when p_n exists, it may be arranged as a Newton polynomial; so (1.2.2) is equivalent to definition (1.1.2).

---

†For example Lagrange's interpolating formula $p_n(\lambda) = \sum_{k=0}^{n} l_k(\lambda) \cdot f(\lambda_k)$, where for each k $l_k(\lambda) \equiv \prod_{j=0}^{n} (\lambda - \lambda_j) / \prod_{j \neq k} (\lambda_k - \lambda_j)$, is one of the simplest. Here $l_k(\lambda_i) = 0$ when $k \neq i$, and $l_k(\lambda_k) = 1$.

$$A = \begin{vmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 2 \end{vmatrix} \qquad B = \begin{vmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & \\ & & & & 2 \end{vmatrix}$$

$$\mu_A(\lambda) = (\lambda-1)(\lambda-2) \qquad \mu_B(\lambda) = (\lambda-1)^4(\lambda-2)$$

$$\chi_A(\lambda) = (\lambda-1)^4(\lambda-2) \qquad \chi_B(\lambda) = \mu_B(\lambda)$$

$$p_m(A) = f(1)I + [f(2)-f(1)](A-I) \qquad p_m(B) = \sum_{k=0}^{3} \frac{f^{(k)}(1)}{k!}(B-I)^k + \Delta_0^4 f\,(B-I)^4$$

$$p_n(A) = \sum_{k=0}^{3} \frac{f^{(k)}(1)}{k!}(A-I)^k + \Delta_0^4 f\,(A-I)^4 \qquad p_n(B) = p_m(B)$$

Fig. 1.2.1: p_m depends on the eigenspaces of the matrix.

The Newton polynomial representation of $f(A)$ requires no more of f than that it have enough derivatives to define $f(\Lambda_A)$. Our interest here, however, concerns functions f holomorphic on a region containing Λ_A. In such cases there is a natural extension of the Newton polynomial to a series. Such a series may be viewed as an interpolating polynomial of infinite order.

This extension derives from a Newton divided difference series,

$$f(\lambda) = \sum_{k=0}^{\infty} \Delta_0^k f \cdot \prod_{j=0}^{k-1} (\lambda - \mu_j)\,, \tag{1.2.3}$$

where the divided differences of f are defined on a sequence of expansion points $\mathbf{M} \equiv \{\mu_0, \mu_1, \mu_2, \ldots\}$ which lies in the domain of holomorphy of f. Because (1.2.3) may be unfamiliar, Appendix A.1 presents an elementary proof demonstrating its convergence. Appendix A.2 establishes the following representation of $f(A)$. Gantmacher [1959] establishes more general series representations for $f(A)$, and Gel'fond [1971] discusses more complicated divided difference expansions. These more extensive results are not needed here.

Newton series representation of f(A). Let f have the Newton divided difference expansion (1.2.3) on an open disk containing Λ_A. Then

$$f(A) = \sum_{k=0}^{\infty} \Delta_0^k f \cdot \prod_{j=0}^{k-1} (A - \mu_j I). \tag{1.2.4}$$

When the first $n+1$ elements of M comprise Λ_A, i.e. $\mu_0 = \lambda_0, \ldots, \mu_n = \lambda_n$, the Newton expansion of $f(A)$ (1.2.4) terminates after the n-th term and is just the Newton polynomial (1.2.2). This is the Cayley-Hamilton theorem, $\prod_{j=0}^{n} (A - \lambda_j I) = \chi_A(A) = 0$.

When $\mathbf{M} = \{\mu, \mu, \mu, \ldots\}$ consists of one point, then each $\Delta_0^k f = f^{(k)}(\mu)/k!$ (§2.1). The Newton expansion (1.2.3) is, then, just a Taylor series; the representation (1.2.4) reduces to a Taylor series for $f(A)$.

Taylor series representation of f(A). Let f have a Taylor series on an open disk about μ containing Λ_A. Then

$$f(A) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\mu)}{k!} (A - \mu I)^k. \tag{1.2.5}$$

The above shows that $f(A)$ is representable in terms of f's divided differences. In the next chapter we reverse this situation. Divided differences of f are expressed in terms of a function of a special matrix. Hence everything said here concerning $f(A)$ applies to divided differences of f, and techniques suitable for computing $f(A)$ may be applied to compute them. In turn, these differences may be used to compute $f(A)$ by the Newton polynomial.

2. Divided Differences

2.1 Definitions and properties of divided differences.

Divided differences were studied extensively in classical precomputer numerical analysis as part of a finite difference calculus. They primarily saw use in tabulation of tables of function values. A quite different purpose is envisioned here; however, much of the classical theory is still relevant. Before proceeding to develop formulas for the calculation of divided differences, we present a few well-known definitions and their consequences. Our notation is somewhat different from that of other authors, but it is felt to be an improvement. Once understood, it will cause no confusion to those already familiar with divided differences.

Most common notations for divided differences are cumbersome. For clarity we begin with such a notation, but later reduce it to more compact form by suppressing unneeded information. Let f be a function of a single variable ζ and be defined, at least, on a sequence $\mathbf{Z} = \{\zeta_0, \zeta_1, \ldots, \zeta_n, \ldots\}$ of distinct complex numbers. $\mathbf{Z}$ is called the sequence of abscissae, or sometimes the sequence of data points or nodes. The 0-th divided difference of f at ζ_0 is

$$(\blacktriangle^0 f)(\zeta_0) \equiv f(\zeta_0).$$

The first divided difference of f at ζ_0 is a function of the two variables (abscissae) ζ_0 and ζ_1; it is formed from the 0-th divided difference by the familiar formula

$$(\blacktriangle^1 f)(\zeta_0, \zeta_1) \equiv \frac{(\blacktriangle^0 f)(\zeta_1) - (\blacktriangle^0 f)(\zeta_0)}{\zeta_1 - \zeta_0} = \frac{f(\zeta_1) - f(\zeta_0)}{\zeta_1 - \zeta_0}.$$

The k-th order divided difference of f at ζ_0 is, then, a function of the $k+1$ abscissae $\zeta_0, \zeta_1, \ldots, \zeta_k$, and is defined iteratively from $k-1$-st order divided differences.

A first definition of divided differences. When f is defined on $\mathbf{Z}$, each k-th order divided difference of f at ζ_j, $j = 0, 1, \ldots, n-k$, is

$$(\blacktriangle^k f)(\zeta_j, \zeta_{j+1}, \ldots, \zeta_{j+k}) \equiv \frac{(\blacktriangle^{k-1} f)(\zeta_{j+1}, \ldots, \zeta_{j+k}) - (\blacktriangle^{k-1} f)(\zeta_j, \ldots, \zeta_{j+k-1})}{\zeta_{j+k} - \zeta_j}. \quad (2.1.1)$$

$(\blacktriangle^k f)(\zeta_j, \zeta_{j+1}, \ldots, \zeta_{j+k})$ has no dependence on abscissae with indices $< j$ or $> j+k$, and so no generality is lost when considering just $(\blacktriangle^n f)(\zeta_0, \zeta_1, \ldots, \zeta_n)$.

Divided differences are very special functions of the data points in Z. Not only does the number of data points used increase with the order of the difference, but the divided difference is symmetric in its arguments. This is obvious in the equivalent representation of the divided difference in terms of determinants [Milne-Thomson, 1933].

$$(\Delta^n f)(\zeta_0, \ldots, \zeta_n) = \begin{vmatrix} f(\zeta_0) & f(\zeta_1) & \cdot & f(\zeta_n) \\ \zeta_0^{n-1} & \zeta_1^{n-1} & \cdot & \zeta_n^{n-1} \\ \cdot & \cdot & & \cdot \\ 1 & 1 & \cdot & 1 \end{vmatrix} \div \begin{vmatrix} \zeta_0^n & \zeta_1^n & \cdot & \zeta_n^n \\ \zeta_0^{n-1} & \zeta_1^{n-1} & \cdot & \zeta_n^{n-1} \\ \cdot & \cdot & & \cdot \\ 1 & 1 & \cdot & 1 \end{vmatrix} \tag{2.1.2}$$

The abscissae may be arranged in any order without changing the value of $(\Delta^n f)(\zeta_0, \zeta_1, \ldots, \zeta_n)$.

Symmetry property. Let π be a permutation on the set of indices $0, 1, \ldots, n$. Then

$$(\Delta^n f)(\zeta_0, \zeta_1, \ldots, \zeta_n) = (\Delta^n f)(\zeta_{\pi(0)}, \zeta_{\pi(1)}, \ldots, \zeta_{\pi(n)}). \tag{2.1.3}$$

When f is symmetric in the real axis, i.e. $f(\bar{\zeta}) = \overline{f(\zeta)}$, (2.1.2) leads to a conjugate symmetry. For odd values of n, $(\Delta^n f)(\zeta_0, \zeta_1, \ldots, \zeta_n)$ is real whenever $\zeta_{2i+1} = \bar{\zeta}_{2i}$, $i = 0, 1, \ldots, (n-1)/2$. And for n even, $(\Delta^n f)(\zeta_0, \zeta_1, \ldots, \bar{\zeta}_n)$ is the conjugate of $(\Delta^n f)(\zeta_0, \zeta_1, \ldots, \zeta_n)$ when each $\zeta_{2i+1} = \bar{\zeta}_{2i}$, $i = 0, 1, \ldots, (n-2)/2$.

The defect in definition (2.1.1) is that data points must be distinct. However when f is differentiable, (2.1.1) may still be defined even for confluent (i.e. equal) abscissae. In particular when $Z = \{\zeta_0, \zeta_0, \ldots, \zeta_0\}$, (2.1.1) is defined when $f^{(n)}(\zeta_0)$ exists. For confluent abscissae the divided difference reduces to

$$(\Delta^n f)(\zeta_0, \zeta_0, \ldots, \zeta_0) = \frac{f^{(n)}(\zeta_0)}{n!}. \tag{2.1.4}$$

Since the data points may be arranged in any order without changing the value of the divided difference, (2.1.1) is defined when (2.1.4) is used when confluent abscissae occur. The requirement that the abscissae be distinct may be removed.

Definition: Let $Z \equiv \{\zeta_0, \ldots, \zeta_0, \zeta_1, \ldots, \zeta_1, \ldots, \zeta_l, \ldots, \zeta_l, \ldots\}$ be a sequence of abscissae (just a renumbering of the previous Z) where each ζ_i, $i = 0, 1, \ldots, l$, appears n_i+1 times, $\sum_{i=0}^{l}(n_i+1) = n+1$. The function f is "defined on the sequence of abscissae Z" when $f(\zeta_i)$,

$f'(\zeta_i), \ldots, f^{(m_i)}(\zeta_i)$ are defined for each $i = 0, 1, \ldots, l$. This sequence of values is denoted $f(\mathbf{Z})$.

Before rewriting definition (2.1.1) in more generality, we introduce a compact notation. The sequence of data points $\mathbf{Z} = \{\zeta_0, \zeta_1, \ldots, \zeta_n, \ldots\}$ is given and, usually, in a fixed order. Hence reference to $\mathbf{Z}$ may be suppressed. Thus we define

$$\Delta_j^k f \equiv (\Delta^k f)(\zeta_j, \zeta_{j+1}, \ldots, \zeta_{j+k}). \tag{2.1.5}$$

The subscript j is understood to mean that we locate the abscissa labled ζ_j and use it and the next k abscissae in the sequence. In the event that the particular sequence $\mathbf{Z}$ must be emphasized, $\Delta_{\zeta_j}^k f$ will be written for $\Delta_j^k f$.†

Standard iterative divided difference scheme. When f is defined on the sequence of abscissae $\mathbf{Z}$,

$$\Delta_j^k f \equiv \frac{\Delta_{j+1}^{k-1} f - \Delta_j^{k-1} f}{\zeta_{j+k} - \zeta_j} \tag{2.1.6}$$

for each $k = 1, 2, \ldots, n$ and $j = 0, 1, \ldots, n-k$, where $\Delta_j^0 f \equiv f(\zeta_j)$.

This definition of divided differences and our earlier definition of matrix functions in §1.1 are consistent. Indeed when $\mathbf{Z} = \Lambda_A$, "defined on the sequence of abscissae $\mathbf{Z}$" and "defined on the characteristic values of A" are the same. We shall see later in §2.6 that this similarity in definitions is no coincidence.

Divided differences have many useful representations and properties. We list several of these here.

Divided difference tables. Divided differences are most conveniently displayed in tables. Traditionally, tables are arranged as in Fig. 2.1.1. Each divided difference is computed from its two immediate neighbors in the column to its left. For our purposes it is most helpful to arrange

†Milne-Thomson [1933] writes $\Delta_0^n f$ as $[\zeta_0, \zeta_1, \ldots, \zeta_n]$, suppressing the function; Davis [1973] uses $f^{[n]}(\zeta_0, \zeta_1, \ldots, \zeta_n)$; and Kahan and Farkas [1963] use $\Delta f(\zeta_0, \zeta_1, \ldots, \zeta_n)$, which suggested the notation used here. Gabel [1968] also uses a similar notation. This compact notation is used in McCurdy [1978], from which much of this introductory section is taken.

$$
\begin{array}{llllll}
\zeta_0 & f(\zeta_0) & & & & \\
 & & \Delta_0^1 f & & & \\
\zeta_1 & f(\zeta_1) & & \Delta_0^2 f & & \\
 & & \Delta_1^1 f & & \Delta_0^3 f & \\
\zeta_2 & f(\zeta_2) & & \Delta_1^2 f & & \Delta_0^4 f \\
 & & \Delta_2^1 f & & \Delta_1^3 f & \\
\zeta_3 & f(\zeta_3) & & \Delta_2^2 f & & \\
 & & \Delta_3^1 f & & & \\
\zeta_4 & f(\zeta_4) & & & &
\end{array}
$$

Fig. 2.1.1: Standard divided difference table.

the table as an upper triangular matrix, for example

$$
\Delta f \equiv \begin{vmatrix}
f(\zeta_0) & \Delta_0^1 f & \Delta_0^2 f & \cdot & \cdot & \cdot & \Delta_0^n f \\
 & f(\zeta_1) & \Delta_1^1 f & \cdot & \cdot & \cdot & \Delta_1^{n-1} f \\
 & & f(\zeta_2) & \cdot & \cdot & \cdot & \Delta_2^{n-2} f \\
 & & & \cdot & & & \cdot \\
 & & & & \cdot & & \cdot \\
 & & & & & \cdot & \cdot \\
 & & & & & & f(\zeta_n)
\end{vmatrix} \tag{2.1.7}
$$

The symbol Δf, without the superscript, is used here to represent a matrix, not a scalar. Elements of the matrix depend on their immediate neighbors in the diagonal to the left. This leads to a "pattern of dependence" in which $\Delta_j^k f$ is independent of all table entries in rows before the j-th and columns after the $j+k$-th. $\Delta_j^k f$ depends only upon the block of the table matrix between it and the main diagonal. Such patterns of dependence are characteristic of triangular matrices.

Linearity. For constants α and β,

$$\Delta_0^n(\alpha f + \beta g) = \alpha \cdot \Delta_0^n f + \beta \cdot \Delta_0^n g \,. \tag{2.1.8}$$

Translation invariance. For $\mathbf{Z}+\mathbf{a} \equiv \{\zeta_0+\alpha, \zeta_1+\alpha, \ldots, \zeta_n+\alpha, \ldots\}$ and $f_\alpha(\zeta) \equiv f(\zeta+\alpha)$,

$$\Delta_{\zeta_0}^n f_\alpha = \Delta_{\zeta_0+\alpha}^n f \,. \tag{2.1.9}$$

For example,

$$\Delta_{\zeta_0}^1 f_\alpha = \frac{f_\alpha(\zeta_1) - f_\alpha(\zeta_0)}{\zeta_1 - \zeta_0} = \frac{f(\zeta_1+\alpha) - f(\zeta_0+\alpha)}{(\zeta_1+\alpha) - (\zeta_0+\alpha)} = \Delta_{\zeta_0+\alpha}^1 f \,.$$

$$
\begin{array}{ccccccccc}
f(\zeta_i) & \rightarrow & \Delta_i^1 f & \rightarrow & \Delta_i^2 f & \rightarrow & \Delta_i^3 f & \rightarrow & \Delta_i^4 f \\
 & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 & & f(\zeta_{i+1}) & \rightarrow & \Delta_{i+1}^1 f & \rightarrow & \Delta_{i+1}^2 f & \rightarrow & \Delta_{i+1}^3 f \\
 & & & & \uparrow & & \uparrow & & \uparrow \\
 & & & & f(\zeta_{i+2}) & \rightarrow & \Delta_{i+2}^1 f & \rightarrow & \Delta_{i+2}^2 f \\
 & & & & & & \uparrow & & \uparrow \\
 & & & & & & f(\zeta_{i+3}) & \rightarrow & \Delta_{i+3}^1 f \\
 & & & & & & & & \uparrow \\
 & & & & & & & & f(\zeta_{i+4})
\end{array}
$$

Fig. 2.1.2: Pattern of dependence in a divided difference table.

Scaling invariance. For some $\tau \neq 0$ let $\tau \mathbf{Z} \equiv \{\tau\zeta_0, \tau\zeta_1, \ldots, \tau\zeta_n, \ldots\}$ and $f_\tau(\zeta) \equiv f(\tau\zeta)$. Then

$$\Delta_{\zeta_0}^n f_\tau = \tau^n \cdot \Delta_{\tau\zeta_0}^n f . \tag{2.1.10}$$

For example,

$$\Delta_{\zeta_0}^1 f_\tau = \frac{f_\tau(\zeta_1) - f_\tau(\zeta_0)}{\zeta_1 - \zeta_0} = \tau \frac{f(\tau\zeta_1) - f(\tau\zeta_0)}{\tau\zeta_1 - \tau\zeta_0} = \tau \cdot \Delta_{\tau\zeta_0}^1 f .$$

$$
\left|\begin{array}{ccccc}
1.000 & 1.718 & 1.476 & .8455 & .3632 \\
 & 2.718 & 4.671 & 4.013 & 2.298 \\
 & & 7.389 & 12.70 & 10.91 \\
 & & & 20.09 & 34.51 \\
 & & & & 54.60
\end{array}\right|
$$

Fig. 2.1.3: Divided difference table for $f = \exp$, with $\mathbf{Z} = \{0, 1, 2, 3, 4\}$.

Mean value representation. When the abscissae are real,

$$\Delta_0^n f = \frac{f^{(n)}(\zeta)}{n!}, \quad \min_{0 \leq j \leq n} \zeta_j \leq \zeta \leq \max_{0 \leq i \leq n} \zeta_i , \tag{2.1.11}$$

for any f having n continuous derivatives in the interval containing the data points. This has no equivalent for complex abscissae. For example when $f = \exp$ and $\zeta_0 = \xi$ and $\zeta_1 = \xi + 2\pi i$,

$$\Delta_0^1 \exp = \frac{e^{\xi + 2\pi i} - e^{\xi}}{(\xi + 2\pi i) - \xi} = 0 \neq e^{\zeta}$$

for any finite ζ.

Integral representation. Another representation for $\Delta_0^n f$, when f has a bounded n-th order derivative on a closed convex domain containing $\mathbf{Z}$, is [Gel'fond, 1971]

$$\Delta_0^n f = \int_0^1 \int_0^{\tau_1} \cdots \int_0^{\tau_{n-1}} f^{(n)}[\zeta_0 + (\zeta_1 - \zeta_0)\tau_1 + \cdots + (\zeta_n - \zeta_{n-1})\tau_n]\, d\tau_n \cdots d\tau_2\, d\tau_1 . \quad (2.1.12)$$

Contour integral representation. When f is holomorphic inside and on a simple closed contour $\mathbf{C}$ enclosing $\mathbf{Z}$, [Gel'fond, 1971]

$$\Delta_0^n f = \frac{1}{2\pi i} \int_{\mathbf{C}} \frac{f(\omega)\, d\omega}{(\omega - \zeta_0)(\omega - \zeta_1) \cdots (\omega - \zeta_n)} . \quad (2.1.13)$$

Bound. If f has a bounded n-th derivative on a closed convex domain $\overline{\Omega}$ containing $\mathbf{Z}$, then [Gel'fond, 1971]

$$|\Delta_0^n f| \leqslant \frac{1}{n!} \max_{\zeta \in \overline{\Omega}} |f^{(n)}(\zeta)| . \quad (2.1.14)$$

This is an immediate consequence of (2.1.12).

In later sections we present a new way of looking at divided differences and develop additional ways to express them.

2.2 A traditional attitude towards divided differences: tables and interpolation.

Before beginning our study of divided differences, we present a short discussion of the traditional attitude towards them in order that contrasts may be made with our approach.

Divided differences are often encountered as an adjunct of the subject of ordinary differences in interpolation and table making. Their treatment in the literature [e.g. Milne-Thomson, 1933, and Miller, 1950] is patterned on that for ordinary differences. The arrangement of the divided difference table (Fig. 2.1.1) is one example. Others are divided difference interpolation formulas which resemble formulas for ordinary differences.† Indeed, our borrowed notation $\blacktriangle$ for the divided difference operator is a modification of Δ for the foreward difference operator.

n	f	Δ	Δ^2	Δ^3	Δ^4	Δ^5
−4						
−3						ϵ
−2					ϵ	
−1				ϵ		-5ϵ
0			ϵ		-4ϵ	
1	ϵ	ϵ		-3ϵ		10ϵ
2		$-\epsilon$	-2ϵ		6ϵ	
3			ϵ	3ϵ		-10ϵ
4				$-\epsilon$	-4ϵ	
					ϵ	5ϵ
						$-\epsilon$

Fig. 2.2.1: Error growth pattern in table of ordinary differences.

The lack of interest in divided differences shown by some authors [e.g. Ralston, 1965] is explicable when we recall their use in interpolation and the available means of computation. In

†Divided difference interpolation formulas are derived from the Newton divided difference interpolating polynomial (1.2.1) in the same way that the Stirling and Bessel formulas are derived from Newton's foreward difference scheme. For example, averaging polynomials using the two sequences of abscissae $\{\zeta_0, \zeta_1, \zeta_{-1}, \zeta_2, \zeta_{-2}, \ldots\}$ and $\{\zeta_0, \zeta_{-1}, \zeta_1, \zeta_{-2}, \zeta_2, \ldots\}$ yields the divided difference generalization of Stirling's formula:

$$p(\zeta) = f(\zeta_0) + \frac{1}{2}\{(\blacktriangle^1 f)(\zeta_0, \zeta_1) + (\blacktriangle^1 f)(\zeta_0, \zeta_{-1})\}\cdot(\zeta - \zeta_0) + (\blacktriangle^2 f)(\zeta_0, \zeta_1, \zeta_{-1})\cdot(\zeta - \zeta_0)(\zeta - \tfrac{\zeta_1 + \zeta_{-1}}{2}) + \ldots .$$

hand or desk-top calculation, we desire few and simple computations. Ordinary difference interpolation formulas may be scaled to minimize divisions and complicated fractions (worse than say $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{12}$, etc.). This is one reason mathematical tables with unevenly spaced data are seldom encountered. Comrie [1959] remarks that "computers try to avoid tabulations at unequal intervals and divided differences ... ."

ζ	Δ^0exp	Δ^1exp	Δ^2exp	Δ^3exp	Δ^4exp	Δ^5exp	Δ^6exp
0.00	1.000						
		0.284					
0.25	1.284		0.081				
		0.365		0.122			
0.50	1.649		0.203		-0.392		
		0.568		-0.270		1.000	
0.75	2.217		-0.067		0.608		-1.997
		0.501		0.338		-0.997	
1.00	2.718		0.271		-0.389		
		0.772		-0.051			
1.25	3.490		0.220				
		0.992					
1.50	4.482						
0.00	0.000						
		0.000					
0.25	0.000		0.000				
		0.000		0.100			
0.50	0.000		0.100		-0.400		
		0.100		-0.300		1.000	
0.75	0.100		-0.200		0.600		-2.000
		-0.100		0.300		-1.000	
1.00	0.000		0.100		-0.400		
		0.000		-0.100			
1.25	0.000		0.000				
		0.000					
1.50	0.000						

Fig. 2.2.2: Example of error propagation in ordinary differences (see Fig. 2.2.5).

The study of error behavior in divided difference computations also shows the domination of ordinary difference theory. For example when an error ϵ occurs only in the $f(0)$ entry, the familiar error growth pattern (Fig. 2.2.1) reveals itself in a table of ordinary differences. The coefficient (exclusive of sign) of the (n, Δ^j) entry is the binomial coefficient

$$\binom{j}{j/2-n}.$$

where n is assigned half values for entries where j is odd.

Miller [1950] writes, "It is also proposed to give error patterns, such as that in [Fig. 2.2.1], for tables of divided differences, for use with tables having certain common arrangements at unequal intervals, for example, with a table having arguments

$$0,\ \frac{1}{4},\ \frac{1}{3},\ \frac{1}{2},\ \frac{2}{3},\ \frac{3}{4},\ 1,\ 1\frac{1}{4},\ 1\frac{1}{3},\ \ldots\ ."$$

Such an error pattern might resemble Fig. 2.2.3.

ζ	f	Δ^1	Δ^2	Δ^3	Δ^4	Δ^5
ζ_{-4}						
ζ_{-3}						$\epsilon_{-5,5}$
ζ_{-2}					$\epsilon_{-4,4}$	$\epsilon_{-4,5}$
ζ_{-1}			$\epsilon_{-2,2}$	$\epsilon_{-3,3}$	$\epsilon_{-3,4}$	$\epsilon_{-3,5}$
ζ_0	ϵ	$\epsilon_{-1,1}$	$\epsilon_{-1,2}$	$\epsilon_{-2,3}$	$\epsilon_{-2,4}$	$\epsilon_{-2,5}$
ζ_1		$\epsilon_{0,1}$	$\epsilon_{0,2}$	$\epsilon_{-1,3}$	$\epsilon_{-1,4}$	$\epsilon_{-1,5}$
ζ_2				$\epsilon_{0,3}$	$\epsilon_{0,4}$	
ζ_3						$\epsilon_{0,5}$
ζ_4						

Fig. 2.2.3: Error growth pattern in table of divided differences.

Each $\epsilon_{i,j} = \epsilon[\prod_{\substack{k=i \\ k\neq 0}}^{i+j}(\zeta_0-\zeta_k)]^{-1}$. The error growth pattern reduces to that of Fig. 2.2.1 when the data points are evenly spaced with unit separation and each entry $\epsilon_{i,j}$ is multiplied by $j!$.

Because n-th order differences of polynomials of degree $n-1$ are zero (see §2.7), one expects high order differences of a function to be small when it is well-approximated by a polynomial. When high order differences begin resembling the alternating sign binomial pattern of Fig. 2.2.1, an error in the tabulated function values is suspected [Miller,1950]. Multiple tabulation errors lead to more complicated patterns, and round-off errors in the difference computations may further obscure any pattern. Statistical methods have been suggested for spotting aberrations in tables [Blanch, 1954].

ζ	Δ^0exp	Δ^1exp	Δ^2exp	Δ^3exp	Δ^4exp	Δ^5exp	Δ^6exp
0.00	1.000						
		1.136					
0.25	1.284		0.6454				
		1.459		1.311			
0.50	1.649		1.629		-4.197		
		2.273		-2.886		8.549	
0.75	2.217		-0.5360		6.489		-11.37
		2.005		3.603		-8.513	
1.00	2.718		2.166		-4.152		
		3.088		-0.5493			
1.25	3.490		1.754				
		3.965					
1.50	4.482						
0.00	0.000						
		0.000					
0.25	0.000		0.000				
		0.000		1.067			
0.50	0.000		0.800		-4.267		
		0.400		-3.200		8.533	
0.75	0.100		-1.600		6.400		-11.38
		-0.400		3.200		-8.533	
1.00	0.000		0.800		-4.267		
		0.000		-1.067			
1.25	0.000		0.000				
		0.000					
1.50	0.000						

Fig. 2.2.4: Example of error propagation in divided differences (see Fig. 2.2.5).

A situation where divided differences, rather than ordinary differences, are usefully employed in interpolation is presented by Salzer [1947]. Bessel functions $J_\nu(\xi)$, $Y_\nu(\xi)$, etc., are commonly tabulated for integral values of ν, as well as $\nu = \pm 1/4$, $\pm 1/3$, $\pm 1/2$, $\pm 2/3$, and $\pm 3/4$.† For ξ fixed, divided differences are used to interpolate for any ν, $-1 \leqslant \nu \leqslant 1$, or to check entries in a table.

Very high (say greater than 10-th) order differences, ordinary or divided, are seldom of practical interest in interpolation problems. The reason is that when the function is tabulated to a fixed number of digits, adjacent table entries often have several initial digits in common.

†See for example tables of the National Bureau of Standards [1948].

Differencing such entries leads to differences containing fewer significant digits. After a few steps no correct digits may remain. The tables in Fig. 2.2.5 illustrate this phenomenon. Less correct information (significant digits) is retained after every differencing step. In interpolation, one is interested only in making a small correction in the last digits of already tabulated values. The information remaining in the first few differences is adequate for this task. However when accurate high order differences are the objects of interest, this loss of information, coupled with magnification of any previously introduced errors when we divide by a small number, is a disaster. We must consider other methods for computing divided differences. The approach necessary to develop such methods forsakes the idea of interpolation between table entries and emphasizes the underlying function.

example: We use the Newton divided difference formula and the four figure divided differences in the second table of Fig. 2.2.5 to interpolate for exp(0.30).

$$\exp(0.30) = 1.000 + 1.136 \times (0.30-0.00) + 0.648 \times (0.30-0.00)(0.30-0.25)$$

$$+ 0.2347 \times (0.30-0.00)(0.30-0.25)(0.30-0.50) + \cdots$$

$$= 1.349816$$

This result correctly interpolates to four figures $\exp(0.30) = 1.350$. The errors in the divided differences do not affect the most significant digits that we want.

Ordinary differences of the exponential

ζ	Δ^0exp	Δ^1exp	Δ^2exp	Δ^3exp	Δ^4exp	Δ^5exp	Δ^6exp
0.00	1.000	0.284	0.081	0.022	0.008	0.000	0.003
0.25	1.284	0.365	0.103	0.030	0.008	0.003	
0.50	1.649	0.468	0.133	0.038	0.011		
0.75	2.117	0.601	0.171	0.049			
1.00	2.718	0.772	0.220				
1.25	3.490	0.992					
1.50	4.482						

Divided differences using 4 digits

ζ	$\blacktriangle^0$exp	$\blacktriangle^1$exp	$\blacktriangle^2$exp	$\blacktriangle^3$exp	$\blacktriangle^4$exp	$\blacktriangle^5$exp	$\blacktriangle^6$exp
0.00	1.000	1.136	6.480E-1	2.347E-1	8.530E-2	0.000	1.712E-2
0.25	1.284	1.460	8.240E-1	3.200E-1	8.530E-2	2.568E-2	
0.50	1.649	1.872	1.064	4.053E-1	1.174E-1		
0.75	2.117	2.404	1.368	5.227E-1			
1.00	2.718	3.088	1.760				
1.25	3.490	3.968					
1.50	4.482						

Correct value of divided differences to 4 digits

ζ	$\blacktriangle^0$exp	$\blacktriangle^1$exp	$\blacktriangle^2$exp	$\blacktriangle^3$exp	$\blacktriangle^4$exp	$\blacktriangle^5$exp	$\blacktriangle^6$exp
0.00	1.000	1.136	6.454E-1	2.444E-1	6.942E-2	1.577E-2	2.987E-3
0.25	1.284	1.459	8.287E-1	3.138E-1	8.913E-2	2.025E-2	
0.50	1.649	1.873	1.064	4.029E-1	1.144E-1		
0.75	2.117	2.405	1.366	5.174E-1			
1.00	2.718	3.088	1.754				
1.25	3.490	3.965					
1.50	4.482						

Fig. 2.2.5: Example of loss of accuracy in computing differences.

2.3 An analytic approach to divided differences.

Up to this point, divided differences are seen as a tool for interpolating in mathematical tables. We assume only that we are given a table of numbers, presumably representing the values of some function at certain arguments. No reference to a particular function or expression is required.

In contrast, we now consider how divided differences depend on the function f and make full use of the theoretical tools presented in §2.1. We treat the divided difference itself as a function; hence, we always assume we can evaluate f and its derivatives at any valid abscissa.† A discussion of tables and interpolation is no longer relevant; neither is a limitation to common arguments, as suggested by Miller [1950] and Salzer [1947]. Indeed, complex as well as real data points are possible. Further, we are interested in floating-point computation on a computer; the desire to avoid divisions, complicated numbers (many digits) and fractions is less important. Finally , we consider divided differences of any order.

example: The power of the analytic approach can be illustrated as follows. We wish to evaluate $\Delta^1\exp$ at the abscissae $\zeta_0=0$ and $\zeta_1=10^{-20}$ on a pocket calculator that can hold only a ten digit number. In the calculator the number $1+10^{-20}$ would be 1.000000000; the 10^{-20} is chopped off. Hence we compute

$$\Delta_0^1\exp = \frac{\exp(10^{-20})-\exp(0)}{10^{-20}-0} \to \frac{1-1}{10^{-20}} = 0$$

as $\exp(10^{-20}) = 1$ to ten digits. Alternatively we may write

$$\Delta_0^1\exp = \frac{\exp(\zeta_1)-\exp(\zeta_0)}{\zeta_1-\zeta_0} = \exp[(\zeta_1+\zeta_0)/2]\cdot\frac{\sinh[(\zeta_1-\zeta_0)/2]}{(\zeta_1-\zeta_0)/2},$$

and then

$$\Delta_0^1\exp = \exp(0.5\times10^{-20})\cdot\frac{\sinh(0.5\times10^{-20})}{0.5\times10^{-20}} \to 1$$

to ten digits, if we can evaluate sinh accurately.

---

†In a computer, abscissae must be representable in the machine; the value of f is rounded (or chopped) at full machine precision.

Divided differences of g using 4 digits							
ζ	$\Delta^0 g$	$\Delta^1 g$	$\Delta^2 g$	$\Delta^3 g$	$\Delta^4 g$	$\Delta^5 g$	$\Delta^6 g$
0.00	0.000	1.406E-6	1.812E-4	2.212E-3	6.913E-3	7.446E-3	2.989E-3
0.25	3.516E-7	9.199E-5	1.840E-3	9.125E-3	1.622E-2	1.193E-2	
0.50	2.335E-5	1.012E-3	8.684E-3	2.534E-2	3.113E-2		
0.75	2.764E-4	5.354E-3	2.769E-2	5.647E-2			
1.00	1.615E-3	1.920E-2	7.004E-2				
1.25	6.416E-3	5.422E-2					
1.50	1.997E-2						

Fig. 2.3.1: Analytic approach to computing $\Delta^6\exp$.

example: Consider the second table in Fig. 2.2.5. The value 0.01712 for $\Delta_0^6\exp$ contains no correct digits. Any sixth order difference of a polynomial of degree five is zero (§2.7). By linearity

$$\Delta_0^6\exp = \Delta_0^6 g\,,$$

where $g = \exp - p_5$ and p_5 is any fifth order polynomial. We set

$$p_5(\zeta) = 1 + \zeta + \frac{\zeta^2}{2!} + \frac{\zeta^3}{3!} + \frac{\zeta^4}{4!} + \frac{\zeta^5}{5!}\,,$$

the first terms of exp's Taylor series. For this choice,

$$g(\zeta) = \sum_{i=6}^{\infty} \frac{\zeta^i}{i!}\,.$$

Using g instead of exp we get $\Delta_0^6\exp = \Delta_0^6 g = 0.002989$, which has three correct decimal digits. The polynomial $p_5(\zeta)$ which dominates the information in the left most digits of $\exp(\zeta)$, the digits lost in differencing, is removed in forming $g(\zeta)$. The information needed to give $\Delta_0^6\exp$ accurately is retained in $g(\zeta)$, but is lost in $\exp(\zeta)$ because too few digits are carried.

There are many cases in which the standard divided difference scheme (2.1.6) works very well (Fig. 2.3.2). Because the scheme is so simple and computationally fast we want to use it, when possible. We need, then, an analysis of the standard formula in order to distinguish those cases where we may wish to employ it. This leads to criteria for deciding when to use it, rather than some other formula.

Divided differences of $\exp_{10}$ by the recursive formula							
ζ	$\Delta^0\exp_{10}$	$\Delta^1\exp_{10}$	$\Delta^2\exp_{10}$	$\Delta^3\exp_{10}$	$\Delta^4\exp_{10}$	$\Delta^5\exp_{10}$	$\Delta^6\exp_{10}$
0.00	1.000	4.472E+1	1.000E+3	1.492E+4	1.668E+5	1.491E+6	1.113E+7
0.25	1.218E+1	5.449E+2	1.219E+4	1.817E+5	2.031E+6	1.818E+7	
0.50	1.484E+2	6.638E+3	1.485E+5	2.213E+6	2.475E+7		
0.75	1.808E+3	8.089E+4	1.808E+6	2.696E+7			
1.00	2.203E+4	9.851E+5	2.203E+7				
1.25	2.683E+5	1.200E+7					
1.50	3.269E+6						

Fig. 2.3.2: Recursive scheme on $\exp_{10}(\zeta) \equiv e^{10\zeta}$, correct $\Delta_0^6\exp_{10} = 1.112E+7$ to 4 digits.

2.4 An error growth analysis of the standard divided difference formula.

The standard divided difference algorithm (2.1.6) propagates, and may magnify, errors introduced at earlier steps.† Algorithms which exhibit this unfortunate error magnification property are often shunned in practice; however, (2.1.6) is just too attractive from the point of view of speed and simplicity to be discarded out of hand. We study here the standard scheme's error behavior and obtain error growth bounds. This analysis provides criteria for deciding when to employ the standard scheme, or another method, to compute $\Delta_0^n f$.

We analyse the error propagation in a typical step of (2.1.6),

$$\Delta_0^n f = \frac{\Delta_1^{n-1} f - \Delta_0^{n-1} f}{\zeta_n - \zeta_0}. \tag{2.4.1}$$

For any expression g let $fl(g)$ represent its computed, or "on hand," value. Employing previously computed values in (2.4.1),

$$fl(\Delta_0^n f) \equiv \frac{fl(\Delta_1^{n-1} f) - fl(\Delta_0^{n-1} f)}{\zeta_n - \zeta_0}.$$

Define

$$fl(\Delta_0^n f) \equiv \Delta_0^n f + \delta_0^n. \tag{2.4.2}$$

δ_0^n is the absolute error in expressing $\Delta_0^n f$ by $fl(\Delta_0^n f)$. Then

$$\Delta_0^n f + \delta_0^n = \frac{(\Delta_1^{n-1} f + \delta_1^{n-1}) - (\Delta_0^{n-1} f + \delta_0^{n-1})}{\zeta_n - \zeta_0}$$

$$= \Delta_0^n f + \frac{\delta_1^{n-1} - \delta_0^{n-1}}{\zeta_n - \zeta_0},$$

and so

$$\delta_0^n = \frac{\delta_1^{n-1} - \delta_0^{n-1}}{\zeta_n - \zeta_0}. \tag{2.4.3}$$

(2.4.3) represents δ_0^n as the error propagated from errors in the $n-1$-st order differences. The

†Algorithms which magnify previously introduced errors from step to step are often referred to as "unstable." This term is commonly applied to algorithms for the numerical solution of differential equations. In this context it is employed, for example, in texts by Richtmeyer and Morton [1967] and Gear [1971].

growth in this propagated error is governed by $(\zeta_n - \zeta_0)^{-1}$, the inverse spread in the abscissae.

We may also define

$$fl(\Delta_0^n f) \equiv \Delta_0^n f \cdot (1 + \epsilon_0^n) . \tag{2.4.4}$$

ϵ_0^n is the relative error in expressing $\Delta_0^n f$ by its computed value $fl(\Delta_0^n f)$. Then

$$\Delta_0^n f \cdot (1 + \epsilon_0^n) = \frac{\Delta_1^{n-1} f \cdot (1+\epsilon_1^{n-1}) - \Delta_0^{n-1} f \cdot (1+\epsilon_0^{n-1})}{\zeta_n - \zeta_0}$$

$$= \Delta_0^n f \cdot [1 + \epsilon_1^{n-1} + \frac{\Delta_0^{n-1} f \cdot (\epsilon_1^{n-1} - \epsilon_0^{n-1})}{\Delta_0^n f \cdot (\zeta_n - \zeta_0)}] ,$$

and so

$$\epsilon_0^n = \epsilon_1^{n-1} + \frac{\Delta_0^{n-1} f \cdot (\epsilon_1^{n-1} - \epsilon_0^{n-1})}{\Delta_0^n f \cdot (\zeta_n - \zeta_0)} \tag{2.4.5}$$

is the relative error in $fl(\Delta_0^n f)$ propagated from relative errors in $n-1$-st order differences.

Expression (2.4.5) indicates the relative error may grow from step to step in (2.1.6), especially when the abscissae ζ_0 and ζ_n are close. This relative error growth is equivalent to the loss of information discussed in §2.2. Such growth in practice may nearly approximate the upper bounds on error growth we derive in §3.2.

Insisting on small relative errors is often inappropriate in divided difference computations. From (2.4.5) one expects a large increase in the relative error when $|\Delta_0^n f|$ is small compared with $|\Delta_0^{n-1} f|$. However a large relative error in a small number is not a disaster when the absolute error is small relative to the final quantity in the computation in which the divided difference is used. Our interest here is accurate computation of n-th order divided differences; we must then, at least, compare the absolute error with an appropriate estimate of the magnitude of n-th order divided differences of f. Conclusions regarding the bounding of the errors expressed in (2.4.3) and, especially, (2.4.5) depend on the particular function f and its own divided differences. We study the exponential function in Chapter 3.

example: Large relative errors in small numbers are not always disastrous. Let $f = \cosh$. Using four digits we compute the following differences. The entry $\Delta_1^1 \cosh = 5.970E{-}3$ has a greater relative error than the other entries; yet subsequent table entries are unaffected by this error because the number 5.970E-3 is

	Table using 4 digits				Correct values to 4 digits			
ζ	$\Delta^0\cosh$	$\Delta^1\cosh$	$\Delta^2\cosh$	$\Delta^3\cosh$	$\Delta^0\cosh$	$\Delta^1\cosh$	$\Delta^2\cosh$	$\Delta^3\cosh$
-2.00	3.762	-2.219	0.7392	0.1638	3.762	-2.219	0.7392	0.1637
-1.00	1.543	0.005970	1.722		1.543	0.005885	1.721	
1.01	1.555	8.614			1.555	8.613		
4.00	27.31				27.31			

Fig. 2.4.1: Large relative errors in small numbers may not be important.

small. In four digit subtraction $0.005970-(-2.219)$ forms $\Delta_0^2\cosh=0.7392$; the incorrect rightmost digits play no role.

2.5 The divided difference as a function of its abscissae.

Our approach to divided differences has not escaped the notion that their formation is an operation performed on entries in a mathematical table. In §2.3 we illustrated the usefulness of an analytic approach. The basis of this approach is that divided differences are functions of their abscissae and may be treated as we treat other mathematical functions.

To aid our discussion we introduce a vector notation for divided differences.† The sequence of abscissae $\mathbf{Z} = \{\zeta_0, \zeta_1, \ldots, \zeta_n\}$ is conveniently viewed as an $n+1$-tuple. Hence $\mathbf{Z}$ is equivalent to a vector $z \equiv (\zeta_0, \zeta_1, \ldots, \zeta_n)$ in $\mathbf{C}^{n+1}$ (or $\mathbf{R}^{n+1}$ for real abscissae). We speak, then, of a divided difference function $\Delta^n f$ being defined for a vector z in the same sense as the function f being "defined on the sequence of abscissae $\mathbf{Z}$" (§2.1). Thus

$$\Delta^n f(z) \equiv (\Delta^n f)(\zeta_0, \zeta_1, \ldots, \zeta_n) . \tag{2.5.1}$$

When every vector z in a region of $\mathbf{C}^{n+1}$ is equivalent to a sequence of abscissae $\mathbf{Z}$ on which f is "defined," $\Delta^n f$ is a function on that region. Thus when defined,

$$\Delta^n f : \mathbf{C}^{n+1} \to \mathbf{C},$$

in brief. Our new notation expresses n-th divided differences of f, $\Delta^n f$, as functions from $\mathbf{C}^{n+1}$ into C. The value of this function at the point $z \in \mathbf{C}^{n+1}$ is $\Delta^n f(z)$. The ordering of the abscissae is suppressed here.

When f is holomorphic on a region containing the abscissae, $\Delta^n f$ is holomorphic in each of its abscissae. In particular for each $i = 0, 1, \ldots, n$,

$$\frac{\partial}{\partial \zeta_i} \Delta^n f(z) = \lim_{\zeta_i' \to \zeta_i} \frac{\Delta^n f(z + (\zeta_i' - \zeta_i)\cdot e_i) - \Delta^n f(z)}{\zeta_i' - \zeta_i}$$

$$= \Delta^{n+1} f(z, \zeta_i) . \tag{2.5.2}$$

The vector $e_i \equiv (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ is the i-th coordinate vector in $\mathbf{C}^{n+1}$. The partial derivative with respect to ζ_i of $\Delta^n f$ is an $n+1$-st order divided difference with the abscissa ζ_i repeated; this is indicated by (z, ζ_i).

†The first notation $(\Delta^n f)(\zeta_0, \zeta_1, \ldots, \zeta_n)$ emphasizes both the sequence of abscissae and its ordering. The second notation $\Delta_0^n f$ merely emphasizes the sequence ordering; it suppresses reference to a particular sequence.

That $\Delta^n f$ is holomorphic in each of its abscissae separately suggests it may be expanded in a series. The next three sections develop such expansions.

example: By differentiating successive divided differences we can show that

$$(\Delta^n \exp)(0, \zeta, \zeta, \ldots, \zeta) = e^{\zeta} \cdot \sum_{k=n}^{\infty} \frac{(-\zeta)^{k-n}}{k!}.$$

Start with

$$(\Delta^1 \exp)(0, \zeta) = \frac{e^{\zeta} - 1}{\zeta} = e^{\zeta} \frac{1 - e^{-\zeta}}{\zeta} = e^{\zeta} \cdot \sum_{k=1}^{\infty} \frac{(-\zeta)^{k-1}}{k!}.$$

With $\zeta_1 = \zeta$,

$$(\Delta^2 \exp)(0, \zeta, \zeta) = \frac{\partial}{\partial \zeta_1} (\Delta^1 \exp)(0, \zeta_1) = \frac{d}{d\zeta} (\Delta^1 \exp)(0, \zeta)$$

$$= e^{\zeta} \{ \sum_{l=1}^{\infty} \frac{(-\zeta)^{l-1}}{l!} - \sum_{k=2}^{\infty} (k-1) \frac{(-\zeta)^{k-2}}{k!} \}$$

$$= e^{\zeta} \cdot \sum_{k=2}^{\infty} \{ \frac{(-\zeta)^{k-2}}{(k-1)!} - (k-1) \frac{(-\zeta)^{k-2}}{k!} \}$$

$$= e^{\zeta} \cdot \sum_{k=2}^{\infty} \frac{(-\zeta)^{k-2}}{k!}.$$

Using the chain rule in the general case,

$$(\Delta^{n+1} \exp)(0, \zeta, \ldots, \zeta) = \frac{\partial}{\partial \zeta_1} (\Delta^n \exp)(0, \zeta_1, \zeta, \ldots, \zeta) = \frac{1}{n} \cdot \frac{d}{d\zeta} (\Delta^n \exp)(0, \zeta, \ldots, \zeta)$$

$$= \frac{1}{n} e^{\zeta} \{ \sum_{l=n}^{\infty} \frac{(-\zeta)^{l-n}}{l!} - \sum_{k=n+1}^{\infty} (k-n) \frac{(-\zeta)^{k-n-1}}{k!} \}$$

$$= \frac{1}{n} e^{\zeta} \cdot \sum_{k=n+1}^{\infty} \{ \frac{(-\zeta)^{k-n-1}}{(k-1)!} - (k-n) \frac{(-\zeta)^{k-n-1}}{k!} \}$$

$$= e^{\zeta} \cdot \sum_{k=n+1}^{\infty} \frac{(-\zeta)^{k-n-1}}{k!}.$$

This may be compared with a method based upon the standard formula,

$$(\Delta^{n+1} \exp)(0, \zeta, \zeta, \ldots, \zeta) = \frac{(\Delta^n \exp)(\zeta, \zeta, \ldots, \zeta) - (\Delta^n \exp)(0, \zeta, \ldots, \zeta)}{\zeta}$$

$$= \frac{e^{\zeta}}{\zeta}\left(\frac{1}{n!} - \sum_{k=n}^{\infty}\frac{(-\zeta)^{k-n}}{k!}\right)$$

$$= e^{\zeta}\cdot\sum_{k=n+1}^{\infty}\frac{(-\zeta)^{k-n-1}}{k!}.$$

example: Two-point divided differences. Just as divided differences for confluent abscissae (which may be referred to as one-point divided differences) reduce to a special form, divided differences about two repeated abscissae also have special properties. Let the vector $z = (\zeta, -\zeta, \ldots, \zeta, -\zeta)$ consist of n repetitions of the two data points ζ and $-\zeta$. Recalling the contour integral formula for divided differences (2.1.13), for $n = 0, 1, 2, \ldots$

$$\Delta^{2n} f(z,\zeta) = \frac{1}{2\pi i}\int_C \frac{f(\omega)\,d\omega}{(\omega-\zeta)^{n+1}(\omega+\zeta)^n},$$

$$\Delta^{2n} f(z,-\zeta) = \frac{1}{2\pi i}\int_C \frac{f(\omega)\,d\omega}{(\omega-\zeta)^{n}(\omega+\zeta)^{n+1}},$$

and

$$\Delta^{2n+1} f(z,\zeta,-\zeta) = \frac{1}{2\pi i}\int_C \frac{f(\omega)\,d\omega}{(\omega-\zeta)^{n+1}(\omega+\zeta)^{n+1}}.$$

The first $2n$ abscissae are represented by z for compactness. For each n define the functions b_n and a_n by

$$b_n(\zeta) \equiv \frac{1}{2}\{\Delta^{2n} f(z,\zeta) + \Delta^{2n} f(z,-\zeta)\} = \frac{1}{2\pi i}\int_C \frac{\omega f(\omega)\,d\omega}{(\omega-\zeta)^{n+1}(\omega+\zeta)^{n+1}}$$

and

$$a_n(\zeta) \equiv \Delta^{2n+1} f(z,\zeta,-\zeta) = \Delta^{2n+1} f(z,-\zeta,\zeta) = \frac{1}{2\pi i}\int_C \frac{f(\omega)\,d\omega}{(\omega-\zeta)^{n+1}(\omega+\zeta)^{n+1}}.$$

These functions are holomorphic in ζ,

$$\frac{d}{d\zeta} b_n(\zeta) = \frac{1}{2\pi i}\cdot\frac{d}{d\zeta}\int_C \frac{\omega f(\omega)\,d\omega}{(\omega-\zeta)^{n+1}(\omega+\zeta)^{n+1}}$$

$$= \frac{2(n+1)\zeta}{2\pi i}\int_C \frac{\omega f(\omega)\,d\omega}{(\omega-\zeta)^{n+2}(\omega+\zeta)^{n+2}}$$

$$= 2(n+1)\zeta \cdot b_{n+1}(\zeta) .$$

and similarly

$$\frac{d}{d\zeta} a_n(\zeta) = 2(n+1)\zeta \cdot a_{n+1}(\zeta) .$$

The functions, then, satisfy the recurrences

$$b_n(\zeta) = \frac{1}{2n\zeta} \cdot \frac{d}{d\zeta} b_{n-1}(\zeta)$$

$$a_n(\zeta) = \frac{1}{2n\zeta} \cdot \frac{d}{d\zeta} a_{n-1}(\zeta)$$

and can be defined when we know $b_0(\zeta)$ and $a_0(\zeta)$. For example when $f = \exp_\tau$, $\exp_\tau(\zeta) \equiv e^{\tau\zeta}$ with $\tau \geqslant 0$,

$$b_0(\zeta) = \cosh(\tau\zeta)$$

$$a_0(\zeta) = \sinh(\tau\zeta)/\zeta .$$

Since

$$a_n(\zeta) = \blacktriangle^{2n+1} f(z, \zeta, -\zeta) = \frac{\blacktriangle^{2n} f(z, \zeta) - \blacktriangle^{2n} f(z, -\zeta)}{2\zeta} ,$$

the divided differences are recoverable from $b_n(\zeta)$ and $a_n(\zeta)$. The values $b_n(\zeta)$ and $a_n(\zeta)$ yield coefficients of a Newton expansion of f about ζ and $-\zeta$. Note that when $\zeta = i\eta$ is pure imaginary, both $b_n(\zeta)$ and $a_n(\zeta)$ are real for any f such that $f(\bar{\zeta}) = \overline{f(\zeta)}$. We extend this example in §2.8.

2.6 The divided difference table as a function of a matrix.

The entire divided difference table (§2.1) of f for the sequence of abscissae $\mathbf{Z} = (\zeta_0, \zeta_1, \ldots, \zeta_n)$ may be expressed as an $(n+1) \times (n+1)$ upper triangular matrix†

$$\Delta f \equiv \begin{vmatrix} f(\zeta_0) & \Delta_0^1 f & \Delta_0^2 f & \cdot & \cdot & \cdot & \Delta_0^n f \\ & f(\zeta_1) & \Delta_1^1 f & \cdot & \cdot & \cdot & \Delta_1^{n-1} f \\ & & f(\zeta_2) & \cdot & \cdot & \cdot & \Delta_2^{n-2} f \\ & & & \cdot & & & \cdot \\ & & & & \cdot & & \cdot \\ & & & & & \cdot & \cdot \\ & & & & & & f(\zeta_n) \end{vmatrix} \tag{2.6.1}$$

Let Z be the special $(n+1) \times (n+1)$ bidiagonal matrix

$$Z = \begin{vmatrix} \zeta_0 & 1 & & & & & \\ & \zeta_1 & 1 & & & & \\ & & \zeta_2 & 1 & & & \\ & & & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \\ & & & & & \zeta_{n-1} & 1 \\ & & & & & & \zeta_n \end{vmatrix} \tag{2.6.2}$$

Opitz [1964] refers to Z as a "steigungsmatrix" (ascent matrix). We shall call Z a "step matrix."

The same conditions on f imply the existence of both the divided difference table Δf (§2.1) and the Newton polynomial representation of $f(Z)$ (§1.2). The two are related as follows.

Theorem: "The divided difference table is a matrix function."

$$\Delta f = f(Z) \tag{2.6.3}$$

proof: The Newton polynomial representation of $f(Z)$ is

$$f(Z) = f(\zeta_0) \cdot I + \Delta_0^1 f \cdot (Z - \zeta_0 I) + \cdots + \Delta_0^n f \cdot \prod_{\lambda=0}^{n-1} (Z - \zeta_\lambda I) .$$

† $\Delta_Z f$ is written when the sequence of abscissae $\mathbf{Z}$ must be emphasized. Recall that Δf, no superscript, is a matrix and $\Delta^n f$ is a scalar function.

Because $Z-\zeta_k I$, each k, is bidiagonal, the product matrices $\prod_{k=0}^{m}(Z-\zeta_k I)$ for $m < n-1$ have $(0,n)$ element zero, while the $(0,n)$ element of $\prod_{k=0}^{n-1}(Z-\zeta_k I)$ is one. Thus the $(0,n)$ element of $f(Z)$ is $\Delta_0^n f$, the $(0,n)$ element of Δf. By the pattern of dependence (§2.1), the choice of 0-th and n-th abscissae is arbitrary. Hence equality holds between every element of $f(Z)$ and Δf. □

Parlett's recurrence (1.1.5) reduces to the standard divided difference scheme (2.1.6) when the upper triangular matrix T is replaced by the step matrix Z. This provides another way to establish (2.6.3).

Several important and useful consequences follow from the theorem.

1. **Function of a Jordan block.** When the sequence of abscissae $\mathbf{Z} = [\zeta_0, \zeta_0, \ldots, \zeta_0]$ is confluent,

$$\Delta f = \begin{vmatrix} f(\zeta_0) & f'(\zeta_0) & \frac{1}{2!}f''(\zeta_0) & \cdot & \cdot & \cdot & \frac{1}{n!}f^{(n)}(\zeta_0) \\ & f(\zeta_0) & f'(\zeta_0) & \cdot & \cdot & \cdot & \frac{1}{(n-1)!}f^{(n-1)}(\zeta_0) \\ & & f(\zeta_0) & \cdot & \cdot & \cdot & \frac{1}{(n-2)!}f^{(n-2)}(\zeta_0) \\ & & & \cdot & & & \cdot \\ & & & & \cdot & & \cdot \\ & & & & & \cdot & \cdot \\ & & & & & & f(\zeta_0) \end{vmatrix} \tag{2.6.4}$$

This is the well-known special form for a function of a Jordan block.

2. **Multiplication formula.** Let the function f_τ be defined by $f_\tau(\zeta) \equiv f(\tau\zeta)$, then

$$\Delta f_\tau = f(\tau Z). \tag{2.6.5}$$

3. **Scaling abscissae.** Let $D \equiv diag(1, \tau, \tau^2, \ldots, \tau^n)$, a diagonal matrix, and $\tau\mathbf{Z} \equiv [\tau\zeta_0, \tau\zeta_1, \ldots, \tau\zeta_n]$, then

$$\Delta_{\mathbf{Z}} f_\tau = D \cdot \Delta_{\tau\mathbf{Z}} f \cdot D^{-1}. \tag{2.6.6}$$

proof: $\Delta_{\mathbf{Z}} f_\tau = f_\tau(Z) = f(\tau Z) = f(DZ_\tau D^{-1}) = D \cdot f(Z_\tau) \cdot D^{-1} = D \cdot \Delta_{\tau\mathbf{Z}} f \cdot D^{-1}$, where

$$Z_\tau \equiv \begin{vmatrix} \tau\zeta_0 & 1 & & & & \\ & \tau\zeta_1 & 1 & & & \\ & & \tau\zeta_2 & 1 & & \\ & & & \cdot & \cdot & \\ & & & & \cdot & \cdot \\ & & & & & \tau\zeta_{n-1} & 1 \\ & & & & & & \tau\zeta_n \end{vmatrix} \quad \square$$

This is the scaling invariance property (2.1.10) in matrix form.

4. **Special functions.** Divided difference tables of certain functions inherit some appealing properties from the functions themselves. For example when $f = \uparrow^j$, the j-th power function† $\uparrow^j(\zeta) \equiv \zeta^j$ for $j = 0, 1, 2, \ldots$,

$$\Delta\uparrow^{j+k} = Z^{j+k} = \Delta\uparrow^j \cdot \Delta\uparrow^k. \tag{2.6.7}$$

Also when $f = \exp_\tau$, $\exp_\tau(\zeta) \equiv e^{\tau\zeta}$,

$$\Delta\exp_{\tau+\sigma} = e^{(\tau+\sigma)Z} = e^{\tau Z} \cdot e^{\sigma Z} = \Delta\exp_\tau \cdot \Delta\exp_\sigma. \tag{2.6.8}$$

†Our divided difference notation suppresses variables, so clarity demands that every function have a name. The notation $\uparrow^j$ for the j-th power function is used by Davis [1973].

2.7 Divided differences of polynomials.

To aid in the development of series expansions for $\Delta''f$, we first examine divided differences of polynomials. Let p_j be a monic polynomial of degree j given in factored form†

$$p_j(\zeta) \equiv \prod_{i=0}^{j-1}(\zeta-\alpha_i) = p_{j-1}(\zeta)\cdot(\zeta-\alpha_{j-1}) \tag{2.7.1}$$

for any $j=0,1,2,\ldots$ $[p_0(\zeta)\equiv 1]$. The polynomial p_j appears in the j-th term of the Newton expansion (§1.2) of f about the sequence $\mathbf{A}\equiv\{\alpha_0,\alpha_1,\alpha_2,\ldots\}$,

$$f = \sum_{j=0}^{\infty}\Delta^j_{\alpha_0}f\cdot p_j\,.$$

p_j reduces to the j-th power function $\uparrow^j$ when all the α_i are zero. For any j and step matrix Z (any sequence of $n+1$ abscissae), the matrix function theorem of the previous section yields

$$\Delta p_j = p_j(Z) \equiv \prod_{i=0}^{j-1}(Z-\alpha_i I)\,. \tag{2.7.2}$$

example: When $n=4$ and $j=3$,

$$\Delta p_3 = p_3(Z) \equiv (Z-\alpha_0 I)\cdot(Z-\alpha_1 I)\cdot(Z-\alpha_2 I)$$

$$= \prod_{i=0}^{2}\begin{vmatrix} \zeta_0-\alpha_i & 1 & & & \\ & \zeta_1-\alpha_i & 1 & & \\ & & \zeta_2-\alpha_i & 1 & \\ & & & \zeta_3-\alpha_i & 1 \\ & & & & \zeta_4-\alpha_i \end{vmatrix}$$

†For a polynomial $p=\sum_{i=0}^{j}\beta_i\uparrow^i$ in non-factored form, the linearity property (2.1.8) yields $\Delta''p=\sum_{i=0}^{j}\beta_i\cdot\Delta''\uparrow^i$.

$$= \begin{vmatrix} (\zeta_0-\alpha_0)(\zeta_0-\alpha_1)(\zeta_0-\alpha_2) & \sum_{k=0}^{1}\sum_{i=0}^{k}(\zeta_k-\alpha_{k+1})(\zeta_i-\alpha_i) & \sum_{i=0}^{2}(\zeta_i-\alpha_i) \\ (\zeta_1-\alpha_0)(\zeta_1-\alpha_1)(\zeta_1-\alpha_2) & \sum_{k=0}^{1}\sum_{i=0}^{k}(\zeta_{k+1}-\alpha_{k+1})(\zeta_{i+1}-\alpha_i) & \\ & (\zeta_2-\alpha_0)(\zeta_2-\alpha_1)(\zeta_2-\alpha_2) & \end{vmatrix}$$

$$\begin{vmatrix} 1 & 0 \\ \sum_{i=0}^{2}(\zeta_{i+1}-\alpha_i) & 1 \\ \sum_{k=0}^{1}\sum_{i=0}^{k}(\zeta_{k+2}-\alpha_{k+1})(\zeta_{i+2}-\alpha_i) & \sum_{i=0}^{2}(\zeta_{i+2}-\alpha_i) \\ (\zeta_3-\alpha_0)(\zeta_3-\alpha_1)(\zeta_3-\alpha_2) & \sum_{k=0}^{1}\sum_{i=0}^{k}(\zeta_{k+3}-\alpha_{k+1})(\zeta_{i+3}-\alpha_i) \\ & (\zeta_4-\alpha_0)(\zeta_4-\alpha_1)(\zeta_4-\alpha_2) \end{vmatrix}$$

In particular, the 0-th (top) row of Δp_3 is

$$\Delta_0^0 p_3 = (\zeta_0-\alpha_0)(\zeta_0-\alpha_1)(\zeta_0-\alpha_2)$$

$$\Delta_0^1 p_3 = (\zeta_0-\alpha_0)(\zeta_0-\alpha_1) + (\zeta_0-\alpha_0)(\zeta_1-\alpha_2) + (\zeta_1-\alpha_1)(\zeta_1-\alpha_2)$$

$$\Delta_0^2 p_3 = (\zeta_0-\alpha_0) + (\zeta_1-\alpha_1) + (\zeta_2-\alpha_2)$$

$$\Delta_0^3 p_3 = 1$$

$$\Delta_0^4 p_3 = 0$$

For general n and j the $(0,n)$ element of Δp_j, that is $\Delta_0^n p_j$, is the $(0,n)$ element of the matrix product $(Z-\alpha_0 I)(Z-\alpha_1 I)\cdots(Z-\alpha_{j-1}I)$. The following formulas can be verified by actually writing out the products. We freely use the convention that $\prod_{i=j}^{k} s_i \equiv 1$ and $\sum_{i=j}^{k} s_i \equiv 0$ when $k < j$, where s_i represents some expression.

Divided differences of polynomials.

$$\Delta_0^n p_j = 0 \text{ when } j < n;$$

$$\Delta_0^n p_n = 1;$$

$$\Delta_0^n p_{n+1} = \sum_{i=0}^{n} (\zeta_i - \alpha_i);$$

$$\Delta_0^n p_{n+2} = \sum_{k=0}^{n} (\zeta_k - \alpha_{k+1}) \sum_{i=0}^{k} (\zeta_i - \alpha_i).$$

In general for all $j \geqslant n$,

$$\Delta_0^n p_j = \prod_{i=0}^{j-n} \{ \sum_{n_i=0}^{n_{i-1}} (\zeta_{n_i} - \alpha_{n_i+j-n-i}) \} \tag{2.7.3}$$

where $n_0 \equiv n$. A rearrangement of this expression is

$$\Delta_0^n p_j = \sum_{\substack{k_0+k_1+\cdots+k_n=j-n \\ k_i \geqslant 0}} \prod_{l=0}^{n} \{ \prod_{l=0}^{k_l-1} (\zeta_l - \alpha_{l+l+k_0+k_1+\cdots+k_{l-1}}) \}. \tag{2.7.4}$$

In the special case of the j-th power function $\uparrow^j$, expression (2.7.4) reduces to

$$\Delta_0^n \uparrow^j = \sum_{\substack{k_0+k_1+\cdots+k_n=j-n \\ k_i \geqslant 0}} \zeta_0^{k_0} \zeta_1^{k_1} \cdots \zeta_n^{k_n}.$$

This is a well-known symmetric polynomial formula for the divided difference of power functions [Milne-Thomson, 1933]. When $n = 1$ the first divided difference of p_j obtained from (2.7.4) is

$$\Delta_0^1 p_j = \sum_{k=0}^{j-1} \{ \prod_{l=0}^{k-1} (\zeta_0 - \alpha_l) \prod_{i=0}^{j-k-2} (\zeta_1 - \alpha_{k+i+1}) \}$$

$$= \sum_{k=0}^{j-1} \{ \prod_{l=0}^{k-1} (\zeta_0 - \alpha_l) \prod_{i=k+1}^{j-1} (\zeta_1 - \alpha_i) \}. \tag{2.7.5}$$

A simple recurrence for computing divided differences of the functions p_{j+1}, $j = -1, 0, 1, 2, \ldots$, may be developed from expression (2.7.2). We begin by writing

$$\Delta p_{j+1} = \Delta p_j \cdot (Z - \alpha_j I). \tag{2.7.6}$$

By writing out the right-hand matrix product, the $(0,k)$ element of Δp_{j+1} is

$$\Delta_0^k p_{j+1} = (\zeta_k - \alpha_j) \cdot \Delta_0^k p_j + \Delta_0^{k-1} p_j \tag{2.7.7}$$

for any $k = 0, 1, \ldots, n$. All the elements $\Delta_0^k p_{j+1}$ for $0 \leqslant k \leqslant n$ comprise the 0-th (i.e. top) row of the matrix Δp_{j+1}.

Formula (2.7.7) is a recurrence in k and j. To see this, we replace the index j in (2.7.7) by $j+k$ where $0 \leqslant k \leqslant n$, and still $j = -1, 0, 1, 2, \ldots$. That is,

$$\Delta_0^k p_{j+k+1} = (\zeta_k - \alpha_{j+k}) \cdot \Delta_0^k p_{j+k} + \Delta_0^{k-1} p_{j+k} \tag{2.7.8}$$

is the $(0,k)$ element of the matrix Δp_{j+k+1}. Thus for fixed j, varying k in (2.7.8) has us looking at elements from the top row of different matrices.

example: Let $j = 2$, then for

$k = 0$, $\Delta_0^k p_{j+k+1} = \Delta_0^0 p_3$ is the $(0,0)$ element of Δp_3;

$k = 1$, $\Delta_0^k p_{j+k+1} = \Delta_0^1 p_4$ is the $(0,1)$ element of Δp_4;

$k = 2$, $\Delta_0^k p_{j+k+1} = \Delta_0^2 p_5$ is the $(0,2)$ element of Δp_5;

and finally for

$k = n$, $\Delta_0^k p_{j+k+1} = \Delta_0^n p_{n+3}$ is the $(0,n)$ element of Δp_{n+3}.

Since $\Delta_0^k p_k = 1$ for all k, the elements $\Delta_0^k p_{j+k+1}$ are known for $j = -1$; so all the $\Delta_0^k p_{k+1}$ are defined by (2.7.8) [we define $\Delta^{-1} p_k \equiv 0$ for any k]. Thus all the $\Delta_0^k p_{j+k+1}$ are computable for $j = 0$, and recursively for any $j > 0$ as well. This procedure is summarized in Algorithm 1, and its first few steps are illustrated in Fig. 2.7.1. Note that if we want all the top row elements of the table Δp_{m+1}, $m \geqslant n$, one element appears in each step of the algorithm from $j = m-n$ to $j = m$. $\Delta_0^n p_{m+1}$ appears first (step $j = m-n$) and $\Delta_0^0 p_{m+1}$ appears last (step $j = m$). Each j-step of the algorithm requires $n+1$ multiplications. Three storage $n+1$-vectors are needed: one to hold the abscissae ζ_i, one to hold the j-th level results $\Delta_0^k p_{k+j+1}$ (the results for level $j-1$ may be overwritten), and one to hold the $n+1$ currently active α_i, namely $\alpha_j, \alpha_{j+1}, \ldots, \alpha_{j+n}$.

Algorithm 1: Recursive computation of $\Delta_0^k p_{k+j}$.

1. Initialize $\Delta_0^k p_k = 1$ for each $k = 0, 1, \ldots, n$
2. For $j = 0, 1, 2, \ldots$

$$\Delta_0^k p_{j+k+1} = (\zeta_k - \alpha_{k+j}) \cdot \Delta_0^k p_{k+j} + \Delta_0^{k-1} p_{k+j}, \quad k = 0, 1, \ldots, n.$$

($\Delta^k f \equiv 0$ for $k < 0$, any function f)

Initialize

$$\Delta_0^0 p_0 = \Delta_0^1 p_1 = \Delta_0^2 p_2 \equiv 1$$

For $j = 0$,

$$\Delta_0^0 p_1 = (\zeta_0 - \alpha_0) \cdot \Delta_0^0 p_0 + \Delta_0^{-1} p_0 = \zeta_0 - \alpha_0$$

$$\Delta_0^1 p_2 = (\zeta_1 - \alpha_1) \cdot \Delta_0^1 p_1 + \Delta_0^0 p_1 = (\zeta_1 - \alpha_1) + (\zeta_0 - \alpha_0)$$

$$\Delta_0^2 p_3 = (\zeta_2 - \alpha_2) \cdot \Delta_0^2 p_2 + \Delta_0^1 p_2 = (\zeta_2 - \alpha_2) + (\zeta_1 - \alpha_1) + (\zeta_0 - \alpha_0)$$

For $j = 1$,

$$\Delta_0^0 p_2 = (\zeta_0 - \alpha_1) \cdot \Delta_0^0 p_1 + \Delta_0^{-1} p_1 = (\zeta_0 - \alpha_1)(\zeta_0 - \alpha_0)$$

$$\Delta_0^1 p_3 = (\zeta_1 - \alpha_2) \cdot \Delta_0^1 p_2 + \Delta_0^0 p_2$$

$$= (\zeta_1 - \alpha_2)(\zeta_1 - \alpha_1) + (\zeta_1 - \alpha_2)(\zeta_0 - \alpha_0) + (\zeta_0 - \alpha_1)(\zeta_0 - \alpha_0)$$

$$\Delta_0^2 p_4 = (\zeta_2 - \alpha_3) \cdot \Delta_0^2 p_3 + \Delta_0^1 p_3$$

$$= (\zeta_2 - \alpha_3)(\zeta_2 - \alpha_2) + (\zeta_2 - \alpha_3)(\zeta_1 - \alpha_1) + (\zeta_2 - \alpha_3)(\zeta_0 - \alpha_0)$$

$$+ (\zeta_1 - \alpha_2)(\zeta_1 - \alpha_1) + (\zeta_1 - \alpha_2)(\zeta_0 - \alpha_0) + (\zeta_0 - \alpha_1)(\zeta_0 - \alpha_0)$$

Fig. 2.7.1: First couple of steps of Algorithm 1 for $n = 2$.

A companion algorithm for computing the n-th column of Δp_{j+1} also exists. Obtaining it merely requires rewriting (2.7.6) as

$$\Delta p_{j+1} = (Z - \alpha_j I)\cdot\prod_{i=0}^{j-1}(Z - \alpha_i I) = (Z - \alpha_j I)\cdot\Delta p_j \tag{2.7.9}$$

and following the same approach as before. Again just one element of the n-th column of a particular matrix is computed at each step. The first few steps of the algorithm are illustrated in Fig. 2.7.2.

Algorithm 2: Recursive computation of $\Delta^{k}_{n-k} p_{k+j}$.

1. Initialize $\Delta^{k}_{n-k} p_k = 1$ for each $k = 0, 1, \ldots, n$
2. For $j = 0, 1, 2, \ldots$

$$\Delta^{k}_{n-k} p_{k+j+1} = (\zeta_{n-k} - \alpha_{k+j})\cdot\Delta^{k}_{n-k} p_{k+j} + \Delta^{k-1}_{n-k+1} p_{k+j}, \quad k = 0, 1, \ldots, n.$$

Initialize

$$\Delta_2^0 p_0 = \Delta_1^1 p_1 = \Delta_0^2 p_2 \equiv 1$$

For $j = 0$,

$$\Delta_2^0 p_1 = (\zeta_2 - \alpha_0)\cdot\Delta_2^0 p_0 + \Delta_3^{-1} p_0 = \zeta_2 - \alpha_0$$

$$\Delta_1^1 p_2 = (\zeta_1 - \alpha_1)\cdot\Delta_1^1 p_1 + \Delta_2^0 p_1 = (\zeta_1 - \alpha_1) + (\zeta_2 - \alpha_0)$$

$$\Delta_0^2 p_3 = (\zeta_0 - \alpha_2)\cdot\Delta_0^2 p_2 + \Delta_1^1 p_2 = (\zeta_0 - \alpha_2) + (\zeta_1 - \alpha_1) + (\zeta_2 - \alpha_0)$$

For $j = 1$,

$$\Delta_2^0 p_2 = (\zeta_2 - \alpha_1)\cdot\Delta_2^0 p_1 + \Delta_3^{-1} p_1 = (\zeta_2 - \alpha_1)(\zeta_2 - \alpha_0)$$

$$\Delta_1^1 p_3 = (\zeta_1 - \alpha_2)\cdot\Delta_1^1 p_2 + \Delta_2^0 p_2$$

$$= (\zeta_1 - \alpha_2)(\zeta_1 - \alpha_1) + (\zeta_1 - \alpha_2)(\zeta_2 - \alpha_0) + (\zeta_2 - \alpha_1)(\zeta_2 - \alpha_0)$$

$$\Delta_0^2 p_4 = (\zeta_0 - \alpha_3)\cdot\Delta_0^2 p_3 + \Delta_1^1 p_3$$

$$= (\zeta_0 - \alpha_3)(\zeta_0 - \alpha_2) + (\zeta_0 - \alpha_3)(\zeta_1 - \alpha_1) + (\zeta_0 - \alpha_3)(\zeta_2 - \alpha_0)$$

$$+ (\zeta_1 - \alpha_2)(\zeta_1 - \alpha_1) + (\zeta_1 - \alpha_2)(\zeta_2 - \alpha_0) + (\zeta_2 - \alpha_1)(\zeta_2 - \alpha_0)$$

Fig. 2.7.2: First couple of steps of Algorithm 2 for $n = 2$.

2.8 Series expansions of $\Delta^n f$.

The matrix function theorem for divided difference tables leads directly to series representations for divided differences. For example suppose that on the disk $\mathbf{D}_\rho = \{\zeta \mid \rho > |\zeta - \alpha|\}$, f has a Newton expansion about the sequence of expansion points $\mathbf{A} = \{\alpha_0, \alpha_1, \alpha_2, \ldots\}$. That is

$$f = \sum_{k=0}^{\infty} \Delta_{\alpha_0}^k f \cdot p_k \tag{2.8.1}$$

on $\mathbf{D}_\rho$, where $p_k(\zeta) = \prod_{j=0}^{k-1} (\zeta - \alpha_j)$. In Appendix A sufficient conditions are presented for the existence of such an expansion. Under the same conditions the matrix function $f(A)$ has a Newton expansion when all the eigenvalues of A lie in $\mathbf{D}_\rho$. Thus when the data points $\mathbf{Z} = \{\zeta_0, \zeta_1, \ldots, \zeta_n\}$ lie in $\mathbf{D}_\rho$, the divided difference table has the Newton expansion

$$\Delta f = f(Z) = \sum_{k=0}^{\infty} \Delta_{\alpha_0}^k f \cdot p_k(Z) \,.^{\dagger} \tag{2.8.2}$$

In the vector notation introduced in §2.5, $\mathbf{Z} \subset \mathbf{D}_\rho$ is equivalent to a vector $z = (\zeta_0, \zeta_1, \ldots, \zeta_n)$ in $\mathbf{D}_\rho^{n+1} \subset \mathbf{C}^{n+1}$. The previous section leads us to examine the $(0,n)$ element of the table. The result is summarized in the following theorem.

Theorem: Newton expansion of the divided difference function. Suppose f has a Newton series on $\mathbf{D}_\rho$. Then

$$\Delta^n f = \sum_{k=n}^{\infty} \Delta_{\alpha_0}^k f \cdot \Delta^n p_k \tag{2.8.3}$$

over all $\mathbf{D}_\rho^{n+1}$, where $p_k(\zeta) = \prod_{j=0}^{k-1} (\zeta - \alpha_j)$.

The important point is the identification by (2.7.2) of the $(0,n)$ element of $p_k(Z)$ with $\Delta_0^n p_k$.

A Taylor series expansion formula for $\Delta^n f$ is an immediate corollary. Recall that $\Delta_{\alpha_0}^k f = f^{(k)}(\alpha)/k!$ in the confluent case.

†The reader is asked to distinguish between the divided differences $\Delta_{\alpha_0}^k f$ forming the series coefficients which have abscissae in A, and the elements $\Delta^n f$ of the divided difference table which have abscissae in Z. Brent [1973] presents a simple Taylor expansion for the divided difference.

Corollary: Taylor expansion of the divided difference function. Suppose f is holomorphic on a region $\mathbf{D}$ containing the point α. Let $\mathbf{D}_\rho = \{\zeta \mid \rho > |\zeta - \alpha|\}$ be the largest open disk such that $\mathbf{D}_\rho \subset \mathbf{D}$. Then

$$\Delta^n f = \sum_{k=n}^{\infty} \frac{f^{(k)}(\alpha)}{k!} \Delta^n \uparrow_\alpha^k \tag{2.8.4}$$

over all $\mathbf{D}_\rho^{n+1}$, where $\uparrow_\alpha^k(\zeta) \equiv \uparrow^k(\zeta - \alpha) = (\zeta - \alpha)^k$.

proof: Because f is holomorphic on $\mathbf{D}_\rho$, it has a Taylor expansion about α for all $\zeta \in \mathbf{D}_\rho$. The theorems in Appendix A establish that

$$f(Z) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\alpha)}{k!} \uparrow^k (Z - \alpha I). \quad \square$$

Formulas (2.8.3) and (2.8.4) suggest ways to compute divided differences for perturbed abscissae when the unperturbed divided differences are available. The computation of divided differences by (2.8.4) for functions such as exp, sin, and cosh is quite straightforward since the Taylor coefficients are easily obtained. Functions such as log and $\sqrt{\ }$ may also be treated; however, care is required to ensure that we use a series representation whose circle of convergence contains all the data points.

The algorithms of §2.7 in combination with (2.8.3) lead to a method for computing $\Delta_0^k f$, $0 \leqslant k \leqslant n$, when we already know the coefficients $\Delta_{\alpha_0}^l f \equiv \beta_l$, $l = 0, 1, 2, \ldots$, of f's Newton expansion. Let $s_m \equiv \sum_{l=0}^{m} \beta_l p_l$ be the partial sums of the Newton expansion (2.8.1) of f, so $s_m \to f$ as $m \to \infty$. Then by linearity

$$\Delta_0^k s_m = \sum_{l=k}^{m} \beta_l \cdot \Delta_0^k p_l ,$$

and by (2.8.3) $\Delta_0^k s_m \to \Delta_0^k f$ as $m \to \infty$ for any k. The following algorithm computes $\Delta_0^k f$ for all $k = 0, 1, \ldots, n$ by forming the partial sums $\Delta_0^k s_m$ for $m = j + k + 1$. One additional term is added to $\Delta_0^k s_{j+k+1}$, for each k, at each j-step.

Algorithm 1: Newton expansion of $\Delta_0^k f$.

1. Initialize $\Delta_0^k p_k = 1$, $\Delta_0^k s_k = \beta_k$, for each $k = 0, 1, \ldots, n$

2. For $j = 0, 1, 2, \ldots$

$$\Delta_0^k p_{j+k+1} = (\zeta_k - \alpha_{j+k}) \cdot \Delta_0^k p_{j+k} + \Delta_0^{k-1} p_{j+k}$$

$$\Delta_0^k s_{j+k+1} = \Delta_0^k s_{j+k} + \beta_{j+k+1} \cdot \Delta_0^k p_{j+k+1}, \quad \text{for each } k = 0, 1, \ldots, n.$$

Exclusive of the coefficient evaluations, the scheme requires $2n+2$ multiplications per j-step.

Initialize

$$\Delta_0^0 p_0 = \Delta_0^1 p_0 = \Delta_0^2 p_0 \equiv 1$$

$$\Delta_0^0 s_0 = \beta_0, \quad \Delta_0^1 s_1 = \beta_1, \quad \Delta_0^2 s_2 = \beta_2$$

For $j = 0$,

$$\Delta_0^0 p_1 = (\zeta_0 - \alpha_0) \cdot \Delta_0^0 p_0 + \Delta_0^{-1} p_0 = \zeta_0 - \alpha_0$$

$$\Delta_0^0 f : \Delta_0^0 s_1 = \Delta_0^0 s_0 + \beta_1 \cdot \Delta_0^0 p_1 = \beta_0 + \beta_1 (\zeta_0 - \alpha_0)$$

$$\Delta_0^1 p_2 = (\zeta_1 - \alpha_1) \cdot \Delta_0^1 p_1 + \Delta_0^0 p_1 = (\zeta_1 - \alpha_1) + (\zeta_0 - \alpha_0)$$

$$\Delta_0^1 f : \Delta_0^1 s_2 = \Delta_0^1 s_1 + \beta_2 \cdot \Delta_0^1 p_2 = \beta_1 + \beta_2 \cdot [(\zeta_1 - \alpha_1) + (\zeta_0 - \alpha_0)]$$

$$\Delta_0^2 p_3 = (\zeta_2 - \alpha_2) \cdot \Delta_0^2 p_2 + \Delta_0^1 p_2 = (\zeta_2 - \alpha_2) + (\zeta_1 - \alpha_1) + (\zeta_0 - \alpha_0)$$

$$\Delta_0^2 f : \Delta_0^2 s_3 = \Delta_0^2 s_2 + \beta_3 \cdot \Delta_0^2 p_3 = \beta_2 + \beta_3 \cdot [(\zeta_2 - \alpha_2) + (\zeta_1 - \alpha_1) + (\zeta_0 - \alpha_0)]$$

Fig. 2.8.1: First step of Algorithm 1 when $n = 2$.

There is also a companion algorithm which computes the n-th column of the matrix Δf.

Algorithm 2: Newton expansion of $\blacktriangle^k_{n-k} f$.

1. Initialize $\blacktriangle^k_{n-k} p_k = 1$, $\blacktriangle^k_{n-k} s_k = \beta_k$, for each $k = 0, 1, \ldots, n$

2. For $j = 0, 1, 2, \ldots$

$$\blacktriangle^k_{n-k} p_{j+k+1} = (\zeta_{n-k} - \alpha_{j+k}) \cdot \blacktriangle^k_{n-k} p_{j+k} + \blacktriangle^{k-1}_{n-k+1} p_{j+k}$$

$$\blacktriangle^k_{n-k} s_{j+k+1} = \blacktriangle^k_{n-k} s_{j+k} + \beta_{j+k+1} \cdot \blacktriangle^k_{n-k} p_{j+k+1}, \quad \text{for each } k = 0, 1, \ldots, n.$$

example: Two-point divided differences (cont. from §2.5). The Newton expansion of f about a sequence of two repeated points $\mathbf{A} = \{\alpha, -\alpha, \alpha, -\alpha, \ldots\}$ is

$$f(\zeta) = f(\alpha) + (\blacktriangle^1 f)(\alpha, -\alpha) \cdot (\zeta - \alpha) + (\blacktriangle^2 f)(\alpha, -\alpha, \alpha) \cdot (\zeta^2 - \alpha^2)$$

$$+ (\blacktriangle^3 f)(\alpha, -\alpha, \alpha, -\alpha) \cdot (\zeta^2 - \alpha^2)(\zeta - \alpha) + \cdots .$$

We may also expand about the rearranged sequence $-\mathbf{A} = \{-\alpha, \alpha, -\alpha, \alpha, \ldots\}$,

$$f(\zeta) = f(-\alpha) + (\blacktriangle^1 f)(-\alpha, \alpha) \cdot (\zeta + \alpha) + (\blacktriangle^2 f)(-\alpha, \alpha, -\alpha) \cdot (\zeta^2 - \alpha^2)$$

$$+ (\blacktriangle^3 f)(-\alpha, \alpha, -\alpha, \alpha) \cdot (\zeta^2 - \alpha^2)(\zeta + \alpha) + \cdots .$$

Recalling the definitions of the functions

$$b_n(\alpha) = \frac{1}{2}\{(\blacktriangle^{2n} f)(\alpha, -\alpha, \ldots, \alpha) + (\blacktriangle^{2n} f)(-\alpha, \alpha, \ldots, -\alpha)\}$$

and

$$a_n(\alpha) = (\blacktriangle^{2n+1} f)(\alpha, -\alpha, \ldots, \alpha, -\alpha)$$

from §2.5, the average of the two expansions is

$$f(\zeta) = b_0(\alpha) + a_0(\alpha) \cdot \zeta + b_1(\alpha) \cdot (\zeta^2 - \alpha^2) + a_1(\alpha) \cdot (\zeta^2 - \alpha^2)\zeta + \cdots .$$

We remarked earlier that when $\alpha = i\eta$ is pure imaginary, $a_n(\alpha)$ and $b_n(\alpha)$ are real for any f such that $f(\bar{\zeta}) = \overline{f(\zeta)}$. Hence the expansion

$$f(\zeta) = b_0(i\eta) + a_0(i\eta) \cdot \zeta + b_1(i\eta) \cdot (\zeta^2 + \eta^2) + a_1(i\eta) \cdot (\zeta^2 + \eta^2)\zeta + \cdots$$

is entirely real when ζ is real.

example: Two-point divided differences of the exponential. When $f = \exp_\tau$, $\exp_\tau(\zeta) \equiv e^{\tau\zeta}$ with $\tau \geq 0$, the coefficients $b_n(\alpha)$ and $a_n(\alpha)$ satisfy the recurrences

$$b_n(\alpha) = \frac{\tau a_{n-1}(\alpha)}{2n}$$

$$a_n(\alpha) = \frac{\tau b_{n-1}(\alpha) - (2n-1)a_{n-1}(\alpha)}{2n\alpha^2}.$$

To show this, recall that (§2.5)

$$b_0(\alpha) = \cosh(\tau\alpha)$$

$$a_0(\alpha) = \sinh(\tau\alpha)/\alpha$$

and

$$b_1(\alpha) = \frac{1}{2\alpha}\cdot\frac{d}{d\alpha}b_0(\alpha) = \frac{\tau\sinh(\tau\alpha)}{2\alpha} = \frac{\tau a_0(\alpha)}{2}$$

$$a_1(\alpha) = \frac{1}{2\alpha}\cdot\frac{d}{d\alpha}a_0(\alpha) = \frac{\tau\cosh(\tau\alpha) - \sinh(\tau\alpha)/\alpha}{2\alpha^2} = \frac{\tau b_0(\alpha) - a_0(\alpha)}{2\alpha^2}.$$

In general (suppressing α for compactness),

$$a_n = \frac{1}{2n\alpha^2}\left\{\frac{\tau a_{n-2}}{2(n-1)} - (2n-1)a_{n-1}\right\}.$$

Differentiating we obtain

$$a'_n = \frac{1}{2n\alpha^2}\left\{\frac{\tau a'_{n-2}}{2(n-1)} - (2n-1)a'_{n-1}\right\} - \frac{2a_n}{\alpha},$$

or

$$2(n+1)\alpha\cdot a_{n+1} = \frac{1}{2n\alpha^2}\{\tau\alpha\cdot a_{n-1} - 2n(2n-1)\alpha\cdot a_n\} - \frac{2a_n}{\alpha}$$

$$= \frac{1}{\alpha}\left\{\frac{\tau a_{n-1}}{2n} - (2n-1)a_n\right\},$$

where we have used that $a'_{n-1} = 2n\alpha\cdot a_n$. Thus

$$a_{n+1} = \frac{1}{2(n+1)\alpha^2}\left\{\frac{\tau a_{n-1}}{2n} - (2n+1)a_n\right\} = \frac{\tau b_n - (2n+1)a_n}{2(n+1)\alpha^2}.$$

This, along with a similar result for b_{n+1}, establishes the recurrences.

For example when $\alpha = 5i$ and $\tau = 1$, we obtain†

$$b_0(5i) = \cos(5) = 2.837E{-}1$$
$$a_0(5i) = \sin(5)/5 = -1.918E{-}1$$
$$b_1(5i) = -9.589E{-}2$$
$$a_1(5i) = -9.509E{-}3$$
$$b_2(5i) = -2.377E{-}3$$
$$a_2(5i) = 6.737E{-}4$$
$$b_3(5i) = 1.123E{-}4$$
$$a_3(5i) = 3.830E{-}5$$
$$b_4(5i) = 4.788E{-}6$$
$$a_4(5i) = 7.792E{-}7.$$

When $\zeta = 1.5\pi i$,

$$\exp(1.5\pi i) = 0.2837 - 0.1918 \times 1.5\pi i - 0.09589 \times [(1.5\pi i)^2 - (5i)^2] + \cdots$$
$$= -1.000i, \text{ as it should.}$$

example: Suppose we want to compute $\Delta_0^2 \exp$ for the slightly perturbed abscissae $\zeta_0 = 5.01i$, $\zeta_1 = -5.01i$ and $\zeta_2 = 4.99i$. Let us follow the steps of Algorithm 1.

Initialize

$$\Delta_0^0 1 = \Delta_0^1(\zeta - 5i) = \Delta_0^1(\zeta + 5i) = \Delta_0^2(\zeta^2 + 25) = 1$$

$$\Delta_0^0 s_0 = 0.2837, \quad \Delta_0^1 s_1 = -0.1918, \quad \Delta_0^2 s_2 = -0.09589$$

For $j = 0$,

$$\Delta_0^0(\zeta - 5i) = (5.01i - 5i) \times 1 = 0.01i$$

$$\Delta_0^0(\zeta + 5i) = (5.01i + 5i) \times 1 = 10.01i$$

$$\Delta_0^0 \zeta = 5.01i$$

$$\Delta_0^0 s_1 = 0.2837 - 0.1918 \times 5.01i = 0.2837 - 0.9608i$$

†Most of the numerical examples here were done on a pocket calculator which, unless a particular working precision is specified, carried more digits than are shown.

$$\Delta_0^1(\zeta^2+25) = (-5.01i+5i)\times 1 + 0.01i = 0$$

$$\Delta_0^1 s_2 = -0.1918$$

$$\Delta_0^2(\zeta^2+25)(\zeta-5i) = (4.99i-5i)\times 1 + 0 = -0.01i$$

$$\Delta_0^2(\zeta^2+25)(\zeta+5i) = (4.99i+5i)\times 1 + 0 = 9.99i$$

$$\Delta_0^2(\zeta^2+25)\zeta = 4.99i$$

$$\Delta_0^2 s_3 = -0.09589 - 0.009509\times 4.99i = -0.09589 - 0.04745i$$

For $j=1$,

$$\Delta_0^0(\zeta^2+25) = (5.01i+5i)\times 0.01i = -0.1001$$

$$\Delta_0^0 s_2 = 0.2837 - 0.9608i - 0.09589\times(-0.1001) = 0.2933 - 0.9608i$$

$$\Delta_0^1(\zeta^2+25)(\zeta-5i) = (-5.01i-5i)\times 0 - 0.1001 = -0.1001$$

$$\Delta_0^1(\zeta^2+25)(\zeta+5i) = (-5.01i+5i)\times 0 - 0.1001 = -0.1001$$

$$\Delta_0^1(\zeta^2+25)\zeta = -0.1001$$

$$\Delta_0^1 s_3 = -0.1918 + 0.009509\times 0.1001 = -0.1908$$

$$\Delta_0^2(\zeta^2+25)^2 = (4.99i+5i)\times(-0.01i) - 0.1001 = -0.0002$$

$$\Delta_0^2 s_4 = -0.09589 - 0.04745i - 0.002377\times(-0.0002) = -0.09589 - 0.04745i$$

The algorithm may be continued for $j=2,3,\ldots$. To four figures, the correct values are

$$\Delta_0^0\text{exp} = 0.2932 - 0.9560i$$

$$\Delta_0^1\text{exp} = -0.1908$$

$$\Delta_0^2\text{exp} = -0.09589 - 0.04745i$$

Note that the conjugate pair divided difference $(\Delta^1\text{exp})(5.01i, -5.01i)$ is real.

2.9 Computing divided difference tables.

We have presented essentially two very different methods for computing divided differences. The first was the standard algorithm (2.1.6) which is simple and fast, but may magnify errors from step to step. The second was the more complicated series algorithm of §2.8.

Because Taylor series coefficients are most easily obtained, the series algorithm is most easily applied for a single expansion point. When the abscissae are closely enough clustered about this expansion point, the series is rapidly convergent. Hence the series algorithm need be computed for only a few steps to obtain divided differences with small error. This is precisely opposite to the case for the standard algorithm, where in §2.4 the error magnification was seen to depend inversely on the separation of the abscissae.

A general purpose algorithm for computing divided difference tables, then, will be a hybrid. Each algorithm above will be used where it is best suited, with primary consideration given to speed and accuracy of computation. The question is then to decide which method to use for a particular element of the table. This is the prime topic of Chapter 3 where divided differences of the exponential function are discussed.

The series algorithm presented in §2.8 computes only one row (or column) of the divided difference table. It could have been written in matrix form in order to give the entire table at once. This is equivalent to applying the given algorithm on each row of the table.

Such repeated applications of the series algorithm is not necessary, however. After one application of the algorithm, the 0-th row of the divided difference table is obtained. We now have sufficient information to fill out the remainder of the table, row by row, by running the standard scheme (2.1.6) backwards.

Backfilling the divided difference table. When divided differences $\Delta_{0,}^{k} f$ for $k = 0, 1, \ldots, n$ are known, the remainder of the table may be obtained by computing successively for $i = 1, 2, \ldots, n$

$$\Delta_{i,}^{k} f = (\zeta_{i+k} - \zeta_{i-1}) \cdot \Delta_{i-1,}^{k+1} f + \Delta_{i-1,}^{k} f, \tag{2.9.1}$$

$k = 0, 1, \ldots, n-i$.

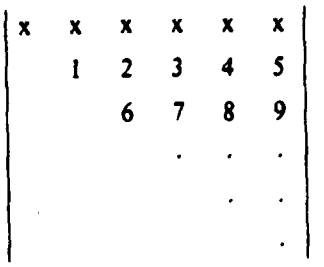


Fig. 2.9.1: Possible order of backfilling using (2.9.1).

As in §2.4, we check how errors are propagated in one step of the backfill algorithm, say to compute $\Delta_1^{n-1} f = \Delta_0^{n-1} f + (\zeta_n - \zeta_0) \cdot \Delta_0^n f$. The absolute error is

$$\delta_1^{n-1} = \delta_0^{n-1} + (\zeta_n - \zeta_0)\delta_0^n; \tag{2.9.2}$$

the relative error is given by

$$\epsilon_1^{n-1} = \frac{\Delta_0^{n-1} f}{\Delta_1^{n-1} f} \epsilon_0^{n-1} + \frac{\Delta_0^n f}{\Delta_1^{n-1} f} (\zeta_n - \zeta_0)\epsilon_0^n. \tag{2.9.3}$$

The absolute error growth is governed by $|\zeta_n - \zeta_0|$. When $\Delta_0^{n-1} f$ and $\Delta_1^{n-1} f$ are of comparable magnitude, the coefficient of ϵ_0^n governs the growth of the relative error. In both (2.9.2) and (2.9.3) the expressions governing error growth are essentially inverses of those governing error growth in (2.4.3) and (2.4.5), respectively. Thus the backfill algorithm is most attractive precisely when the series algorithm is most attractive and the standard scheme is not.

3. Divided Differences of Exponentials

3.1 Special formulas for exponential divided differences.

The ideas presented in the last chapter are illustrated by considering the exponential function. Because the exponential function is entire, all formulas and algorithms discussed so far are applicable for any abscissae. Further, the special properties of the function permit useful simplification of our previous formulas. In addition, results obtained for the exponential may be modified to cover related functions, such as sin or cosh, by means of the linearity property (2.1.8) of divided differences.

The behavior of exponential divided differences under a constant shift in the abscissae illustrates a useful simplification of the translation invariance property (2.1.9). It is convenient to consider the more general function $\exp_\tau$ with scaling parameter τ, that is $\exp_\tau(\zeta) \equiv e^{\tau\zeta}$.

Translation property of exponential divided differences. Let z be a vector whose elements are data points (§2.5). Then for any constant α,

$$\Delta''\exp_\tau(z+\alpha u) = e^{\tau\alpha}\cdot\Delta''\exp_\tau(z) \tag{3.1.1}$$

where the constant vector $u = (1, 1, \ldots, 1)$.

It is clear from (3.1.1) that no generality is lost when we restrict attention to sets of abscissae with, say, $\zeta_0 = 0$ or $\zeta_n = -\zeta_0$. In the latter case the first divided difference simplifies to

$$\Delta_0^1\exp_\tau = \frac{e^{-\tau\zeta_0} - e^{\tau\zeta_0}}{-\zeta_0 - (\zeta_0)} = \frac{\sinh(\tau\zeta_0)}{\zeta_0}.$$

In general for non-centered abscissae any first divided difference of the exponential can be written as

$$\Delta_0^1\exp_\tau = e^{\tau\omega}\frac{\sinh(\tau\psi)}{\psi} \tag{3.1.2}$$

where $\omega \equiv (\zeta_1 + \zeta_0)/2$ and $\psi \equiv (\zeta_1 - \zeta_0)/2$.

The integral representation formula for divided differences (2.1.12) acquires a simpler form when the parameter τ is non-negative. We have

$$\Delta_0^n \exp_\tau = \int_0^1 \int_0^{\tau_1} \cdots \int_0^{\tau_{n-1}} \frac{d^n}{d\zeta^n} \exp_\tau[\zeta_0 + (\zeta_1 - \zeta_0)\tau_1 + \cdots + (\zeta_n - \zeta_{n-1})\tau_n]\, d\tau_n \cdots d\tau_1$$

$$= \int_0^1 \int_0^{\tau_1} \cdots \int_0^{\tau_{n-1}} \tau^n \exp[\tau\zeta_0 + (\zeta_1 - \zeta_0)\tau\tau_1 + \cdots + (\zeta_n - \zeta_{n-1})\tau\tau_n]\, d\tau_n \cdots d\tau_1$$

by the definition of $\exp_\tau$. The change of variables $\sigma_j = \tau\tau_j$ for $j = 1, 2, \ldots, n$ yields the alternative expression

$$\Delta_0^n \exp_\tau = \int_0^\tau \int_0^{\sigma_1} \cdots \int_0^{\sigma_{n-1}} \exp[\tau\zeta_0 + (\zeta_1 - \zeta_0)\sigma_1 + \cdots + (\zeta_n - \zeta_{n-1})\sigma_n]\, d\sigma_n \cdots d\sigma_1 . \quad (3.1.3)$$

We recognize that this is a recurrence for $\Delta_0^n \exp_\tau$, namely

$$\Delta_0^n \exp_\tau = e^{\tau\zeta_0} \int_0^\tau e^{-\sigma\zeta_0} \cdot \Delta_1^{n-1} \exp_\sigma \, d\sigma$$

where $\sigma \equiv \sigma_1$. By the symmetry property (2.1.3), the ordering of the abscissae is arbitrary; we may replace ζ_0 by any ζ_i, $0 \leqslant i \leqslant n$. To deal with such cases we define

$$\Delta_{(i)}^{n-1} \exp_\tau \equiv (\Delta^{n-1} \exp_\tau)(\zeta_0, \zeta_1, \ldots, \zeta_{i-1}, \zeta_{i+1}, \ldots, \zeta_n) , \quad (3.1.4)$$

the $n-1$-st divided difference with the i-th abscissa omitted. (3.1.5) summarizes the formula.

Recursive integral formula for $\Delta_0^n \exp_\tau$. For any $\tau \geqslant 0$ and index $i = 0, 1, \ldots, n$,

$$\Delta_0^n \exp_\tau = e^{\tau\zeta_i} \int_0^\tau e^{-\sigma\zeta_i} \cdot \Delta_{(i)}^{n-1} \exp_\sigma \, d\sigma . \quad (3.1.5)$$

This result will prove useful in the next section. In addition, one recognizes that formula (3.1.5) is a convolution,

$$(\exp_{\zeta_i} * \Delta_{(i)}^{n-1} \exp)(\tau) ,$$

where $\Delta_{(i)}^{n-1} \exp$ is treated as a function in τ. The correspondence is obvious from the convolution formula, with $g(\sigma) = \Delta_{(i)}^{n-1} \exp_\sigma$,

$$(f * g)(\tau) = \int_0^\tau f(\tau - \sigma) \cdot g(\sigma) \, d\sigma .$$

3.2 Real exponential divided differences.

Exponential divided differences for real abscissae are positive and increasing functions of their abscissae. These properties permit derivation of both upper and lower bounds which show how the divided differences and the error growth in the standard formula depend on (*i*) the spread in the abscissae, (*ii*) the order n of the difference, and (*iii*) the parameter τ.

In this and the next four sections we consider exclusively divided differences of the function $f = \exp_\tau$, with parameter $\tau \geq 0$, for real sequences of abscissae $\mathbf{X} \equiv \{\xi_0, \xi_1, \ldots, \xi_n\}$. All such divided differences have two properties which characterize them.

Theorem 1: For all $\tau > 0$ and $n \geq 0$, $\Delta^n \exp_\tau$ is

(*i*) positive,

(*ii*) strictly increasing in each abscissa ξ_i, $i = 0, 1, \ldots, n$.

proof: The result is almost immediate from the recursive integral formula (3.1.5),

$$\Delta_0^n \exp_\tau = e^{\tau\xi_i} \int_0^\tau e^{-\sigma\xi_i} \cdot \Delta_{(i)}^{n-1} \exp_\sigma \, d\sigma \, .$$

All 0-th order real exponential divided differences are positive [$\Delta^0 \exp_\tau(\xi) = e^{\tau\xi}$]. The recurrence implies all first order differences are also positive, and hence by induction all n-th order differences are positive for any n. For (*ii*),

$$\frac{\partial}{\partial \xi_i} \Delta_0^n \exp_\tau = \int_0^\tau (\tau - \sigma) e^{(\tau-\sigma)\xi_i} \cdot \Delta_{(i)}^{n-1} \exp_\sigma \, d\sigma > 0 \, ,$$

since the integrand is positive. □

The recursive integral formula also provides an easy way to develop expressions relating divided differences of orders $n-1$ and n.

Theorem 2: Suppose $\beta \leqslant \xi_i \leqslant \gamma$ for every abscissa ξ_i, $0 \leqslant i \leqslant n$. Then for each i there exists a $\xi \in [\beta, \gamma]$ such that

$$\Delta_{(i)}^{n-1}\exp_\tau = (\xi - \xi_i + \frac{n}{\tau})\cdot\Delta_0^n\exp_\tau . \tag{3.2.1}$$

proof: By the translation (3.1.1) and scaling invariance (2.1.10) properties,

$$\Delta^n\exp[\tau(x-\xi u)] = \tau^{-n}e^{-\tau\xi}\cdot\Delta_0^n\exp_\tau = \tau^{-n}e^{\tau(\xi_i-\xi)}\int_0^\tau e^{-\sigma\xi_i}\cdot\Delta_{(i)}^{n-1}\exp_\sigma\, d\sigma$$

for any $i = 0, 1, \ldots, n$ and ξ. Differentiating with respect to τ yields

$$\frac{d}{d\tau}\Delta^n\exp[\tau(x-\xi u)] = \tau^{-n}e^{-\tau\xi}\{(\xi_i - \xi - \frac{n}{\tau})\cdot\Delta_0^n\exp_\tau + \Delta_{(i)}^{n-1}\exp_\tau\} . \tag{3.2.2}$$

Every element of the vector $x-\beta u$ is non-negative, and so $\Delta^n\exp[\tau(x-\beta u)]$ is increasing in τ. Similarly, every element of $x-\gamma u$ is non-positive and $\Delta^n\exp[\tau(x-\gamma u)]$ is decreasing in τ. Hence

$$\frac{d}{d\tau}\Delta^n\exp[\tau(x-\beta u)] \geqslant 0 \geqslant \frac{d}{d\tau}\Delta^n\exp[\tau(x-\gamma u)] ;$$

so for some $\xi \in [\beta, \gamma]$, the derivative is zero. □

A plethora of upper and lower bounds on $\Delta_0^n\exp_\tau$ can be derived from the simple expression (3.2.1) by choosing particular values of ξ and i. The two simplest follow by choosing ξ as one or the other of the end points of the smallest interval containing the abscissae. Note that equality holds when the abscissae are confluent.

Corollary 1: Lower bound on $\Delta_0^n\exp_\tau$. If $\xi_n \geqslant \xi_i$ for each $i = 0, 1, \ldots, n$, then

$$\Delta_0^n\exp_\tau \geqslant \frac{\tau}{n}\cdot\Delta_0^{n-1}\exp_\tau . \tag{3.2.3}$$

proof: Choose $i = n$ and $\gamma = \xi_n$ in (3.2.1), and note that $\xi - \xi_n \leqslant 0$. □

Corollary 2: Upper bound on $\Delta_0^n \exp_\tau$. If $\xi_0 \leqslant \xi_i$ for each $i = 0, 1, \ldots, n$, then

$$\Delta_0^n \exp_\tau \leqslant \frac{\tau}{n} \cdot \Delta_1^{n-1} \exp_\tau . \tag{3.2.4}$$

(3.2.3) leads directly to a bound on the relative error growth in one step of the standard divided difference algorithm (2.1.6). From (2.4.5) the error in $fl(\Delta_0^n \exp_\tau)$ relative to $\Delta_0^n \exp_\tau$ is

$$\epsilon_0^n = \epsilon_1^{n-1} + \frac{\Delta_0^{n-1} \exp_\tau \cdot (\epsilon_1^{n-1} - \epsilon_0^{n-1})}{\Delta_0^n \exp_\tau \cdot (\xi_n - \xi_0)} .$$

The factor

$$r_0^n(\tau; x) \equiv \frac{\Delta_0^{n-1} \exp_\tau}{\Delta_0^n \exp_\tau \cdot |\xi_n - \xi_0|} , \tag{3.2.5}$$

which we call the growth factor, controls the growth of the relative error in computing $\Delta_0^n \exp_\tau$ by (2.1.6). Clearly when $r_0^n(\tau; x)$ is small, the relative error growth is small; conversely when it is large, the relative error growth may be large. By (3.2.3) the growth factor satisfies

$$r_0^n(\tau; x) \leqslant \frac{n}{\tau(\xi_n - \xi_0)} \tag{3.2.6}$$

when $\xi_n \geqslant \xi_i$ for all i.

(3.2.6) illustrates the dependence of error growth on the three factors mentioned at the beginning of this section. It also permits us to bound propagated errors in the divided difference table in the manner shown in Fig. 2.2.3. We illustrate this in Fig. 3.2.1 for a single initial relative error ϵ in $\Delta_0^0 \exp_\tau$.† Each element of the bottom diagonal satisfies

$$|\epsilon_0^n| \leqslant \tau^{-n}(n!)|\epsilon| \cdot [\prod_{j=1}^{n} (\xi_j - \xi_0)]^{-1} .$$

Note that for equispaced abscissae, say $\xi_j = j\delta$, we have $|\epsilon_0^n| \leqslant |\epsilon|/(\tau\delta)^n$, and the factorial cancels.

†The abscissae are $\cdots \xi_{-4} \leqslant \xi_{-3} \leqslant \xi_{-2} \leqslant \xi_{-1} \leqslant \xi_0 \leqslant \xi_1 \leqslant \xi_2 \leqslant \xi_3 \leqslant \xi_4 \cdots$, in contrast with our usual numbering.

ξ	$\Delta^0\exp_\tau$	$\Delta^1\exp_\tau$	$\Delta^2\exp_\tau$	$\Delta^3\exp_\tau$	$\Delta^4\exp_\tau$	$\Delta^5\exp_\tau$
ξ_{-4}						
ξ_{-3}						ϵ_{-5}^5
ξ_{-2}					ϵ_{-4}^4	
ξ_{-1}			ϵ_{-2}^2	ϵ_{-3}^3		ϵ_{-4}^5
ξ_0	ϵ	ϵ_{-1}^1	ϵ_{-1}^2	ϵ_{-2}^3	ϵ_{-3}^4	ϵ_{-3}^5
ξ_1		ϵ_0^1	ϵ_0^2	ϵ_{-1}^3	ϵ_{-2}^4	ϵ_{-2}^5
ξ_2				ϵ_0^3	ϵ_{-1}^4	ϵ_{-1}^5
ξ_3					ϵ_0^4	ϵ_0^5
ξ_4						

Fig. 3.2.1: Relative error growth pattern in table of real exponential divided differences.

A sharper error growth bound than (3.2.6) is obtainable. We improve (3.2.3) by further refining expression (3.2.2), which was used in the proof of Theorem 2.

Theorem 3: Suppose $\beta \leqslant \xi_i \leqslant \gamma$ for all abscissae ξ_i, $0 \leqslant i \leqslant n$. Then there exist values $\xi'_j \in [\beta, \gamma]$, $0 \leqslant j \leqslant n$, such that for any value of ξ and every index i

$$\Delta_{(i)}^{n-1}\exp_\tau = \{\xi - \xi_i + \frac{n}{\tau} + \sum_{j=0}^{n} \frac{\xi_j - \xi}{n+1+\tau(\xi'_j - \xi_j)}\} \cdot \Delta_0^n \exp_\tau . \tag{3.2.7}$$

proof: By the chain rule for differentiation, (3.1.1), and (2.1.10),

$$\frac{d}{d\tau}\Delta^n \exp[\tau(x-\xi u)] = \sum_{j=0}^{n} \{(\xi_j - \xi)\cdot \Delta^{n+1}\exp[\tau(x-\xi u), \tau(\xi_j - \xi)]\}$$

$$= e^{-\tau\xi}\tau^{-(n+1)} \sum_{j=0}^{n} \{(\xi_j - \xi)\cdot \Delta^{n+1}\exp_\tau(x, \xi_j)\} .$$

Combining this with (3.2.2) yields

$$\Delta_{(i)}^{n-1}\exp_\tau = (\frac{n}{\tau} + \xi - \xi_i)\cdot \Delta_0^n \exp_\tau + \frac{1}{\tau}\sum_{j=0}^{n} \{(\xi_j - \xi)\cdot \Delta^{n+1}\exp_\tau(x, \xi_j)\} .$$

Now by Theorem 2, for each j there exists a $\xi'_j \in [\beta, \gamma]$ such that

$$\Delta^{n+1}\exp_\tau(x,\xi_j) = \frac{\tau}{n+1+\tau(\xi'_j-\xi_j)}\Delta_0^n\exp_\tau. \quad \square$$

Setting $i=n$ and $\xi=\xi'_j=\xi_n$ for all $0\leqslant j\leqslant n$ yields a sharper inequality than (3.2.3) when ξ_n is the largest abscissa.

Corollary: If $\xi_n \geqslant \xi_i$ for each $i=0,1,\ldots,n$, then

$$\Big[\sum_{j=0}^{n-1}\frac{1}{n+1+\tau(\xi_n-\xi_j)}\Big]\cdot\Delta_0^n\exp_\tau \geqslant \frac{\tau}{n+1}\cdot\Delta_0^{n-1}\exp_\tau. \tag{3.2.8}$$

Even better inequalities can be derived, but at the sacrifice of simplicity. We also note that

$$\sum_{j=0}^{n}\frac{1}{n+1+\tau(\xi'_j-\xi_j)} = 1$$

because the left-hand side of (3.2.7) and $\Delta_0^n\exp_\tau$ are independent of ξ, thus giving a relation amongst the ξ'_j.

example: For evenly spaced data points real exponential divided differences can be presented analytically. Let $\mathbf{X}=\{\xi_0,\xi_0+2\delta,\xi_0+4\delta,\ldots,\xi_0+2n\delta\}$, where 2δ is the spacing. Then

$$\Delta_0^n\exp_\tau = \frac{1}{n!}e^{\tau\xi_0}\Big[\frac{e^{2\tau\delta}-1}{2\delta}\Big]^n = \frac{1}{n!}e^{\tau(\xi_0+n\delta)}\Big[\frac{\sinh(\tau\delta)}{\delta}\Big]^n. \tag{3.2.9}$$

This expression yields very accurate divided differences, especially if we have available a good routine to evaluate the function $\mathrm{Sh}(\xi)\equiv\sinh(\xi)/\xi$. Fig. 3.2.2 compares divided differences $\Delta_0^n\exp_\tau$ for $n=0,1,\ldots,24$ computed according to (3.2.9) and the standard algorithm (2.1.6). $\tau=1$, $\xi_0=0$ and $2\delta=1$. The initial values $\Delta^0\exp$ are rounded to seven digits, and all arithmetic is performed in greater precision in order to isolate error growth due to initial errors.

The table in Fig. 3.2.3 compares the actual error growth per step in using the standard scheme with bounds derived from

$$|\epsilon_0^n| \leqslant |\epsilon_1^{n-1}| + r_0^n(\tau;x)\cdot(|\epsilon_1^{n-1}|+|\epsilon_0^{n-1}|)$$

under the assumption $|\epsilon_1^{n-1}|=|\epsilon_0^{n-1}|$. That is, $|\epsilon_0^n|\leqslant\rho\cdot|\epsilon_0^{n-1}|$ where

$$\rho \equiv 1 + 2r_0''(\tau;x).$$

n	$\Delta_0''\exp$ by (3.2.9)	$\Delta_0''\exp$ by (2.1.6)	Relative error in $\Delta_0''\exp$ by (2.1.6)
0	1.000000	1.000000	0.0
1	1.718282	1.718282	0.0
2	1.476247	1.476246	6.774E-7
3	8.455359E-1	8.455363E-1	-4.731E-7
4	3.632173E-1	3.632166E-1	1.927E-6
5	1.248219E-1	1.248225E-1	-4.807E-6
6	3.574655E-2	3.574611E-2	1.231E-5
7	8.774665E-3	8.774811E-3	-1.664E-5
8	1.884669E-3	1.884642E-3	1.433E-5
9	3.598214E-4	3.598186E-4	7.782E-6
10	6.182746E-5	6.183118E-5	-6.017E-5
11	9.657909E-6	9.655845E-6	2.137E-4
12	1.382918E-6	1.383742E-6	-5.958E-4
13	1.827879E-7	1.825252E-7	1.437E-3
14	2.243437E-8	2.249849E-8	-2.858E-3
15	2.569905E-9	2.558588E-9	4.404E-3
16	2.759888E-10	2.772152E-10	-4.444E-3
17	2.789568E-11	2.787636E-11	6.926E-4
18	2.662925E-12	2.645633E-12	6.494E-3
19	2.408240E-13	2.422432E-13	-5.893E-3
20	2.069018E-14	2.148605E-14	-3.847E-2
21	1.692931E-15	1.349106E-15	2.031E-1
22	1.322242E-16	2.119468E-16	-6.029E-1
23	9.878198E-18	-3.198523E-18	1.324
24	7.072304E-19	2.272222E-18	-2.213

Fig. 3.2.2: $\Delta_0''\exp$ computed from initial values rounded to 7 digits.

Error growth factors	p
Average error growth per step	≈ 2
Error growth bound using $r_0''(\tau;x)$	$1+2/(e^1-1)=2.16$
Error growth bound using (3.2.8) to bound $r_0''(\tau;x)$	$<1+2\log 2=2.4$
Error growth bound using (3.2.3) to bound $r_0''(\tau;x)$	3

Fig. 3.2.3: Relative error growth and bounds for divided differences in Fig. 3.2.2.

3.3 The Taylor series algorithm for $\Delta_0^n \exp_\tau$.

The error growth bounds of the last section show that accurate computation of $\Delta_0^n \exp_\tau$ when $\tau|\xi_n - \xi_0|$ is small demands that we consider a method other than the simple scheme (2.1.6). The series algorithm presented in §2.8 suits our requirements. In particular the Taylor coefficients are easy to compute and convergence of the Taylor expansion accelerates as the abscissae cluster more closely together.

The algorithm is derived directly from the Taylor expansion formula (2.8.4). Without loss of generality let $\xi_0 \leqslant \xi_i \leqslant \xi_n$ for each $i = 0, 1, \ldots, n$, and define

$$\alpha \equiv \frac{\xi_n + \xi_0}{2} \quad \text{and} \quad \theta \equiv \xi_n - \xi_0. \tag{3.3.1}$$

α is the Taylor series' expansion point, and θ is the spread in the abscissae. With $f = \exp_\tau$ the basic formula yields

$$\Delta_0^n \exp_\tau = e^{\tau\alpha} \sum_{j=0}^{\infty} \frac{\tau^{n+j}}{(n+j)!} \cdot \Delta_0^n \uparrow_\alpha^{n+j} \tag{3.3.2}$$

where $\uparrow_\alpha^{n+j}(\xi) \equiv \uparrow^{n+j}(\xi - \alpha) = (\xi - \alpha)^{n+j}$. Let

$$s_m \equiv e^{\tau\alpha} \sum_{j=0}^{m} \frac{\tau^j}{j!} \uparrow_\alpha^j$$

be a partial sum of the Taylor series for $\exp_\tau$; so

$$\Delta_0^n s_m = e^{\tau\alpha} \sum_{j=0}^{m-n} \frac{\tau^{n+j}}{(n+j)!} \cdot \Delta_0^n \uparrow_\alpha^{n+j}$$

is a partial sum of $\Delta_0^n \exp_\tau$. Algorithm 1 of §2.8 translates into the following.

Algorithm: Taylor series algorithm for $\Delta_0^j \exp_\tau$.

1. Initialize $\Delta_0^k \uparrow_\alpha^k = 1$, $\Delta_0^k s_k = \dfrac{e^{\tau\alpha}\tau^k}{k!}$, for each $k = 0, 1, \ldots, n$

2. For $j = 0, 1, 2, \ldots$

$$\Delta_0^k \uparrow_\alpha^{j+k+1} = (\xi_k - \alpha) \cdot \Delta_0^k \uparrow_\alpha^{j+k} + \Delta_0^{k-1} \uparrow_\alpha^{j+k}$$

$$\Delta_0^k s_{j+k+1} = \Delta_0^k s_{j+k} + \frac{e^{\tau\alpha}\tau^{j+k+1}}{(j+k+1)!} \cdot \Delta_0^k \uparrow_\alpha^{j+k+1}, \text{ for each } k = 0, 1, \ldots, n.$$

The algorithm produces all the values $\Delta_0^k \exp_\tau$, $0 \leqslant k < n$, as a bonus because the quantities needed to form them are intermediate in forming just $\Delta_0^n \exp_\tau$. Also, the coefficient evaluations can be performed iteratively for greater efficiency.

We now wish to examine how the error in computing $\Delta_0^n \exp_\tau$ by this algorithm is affected by the parameter τ, the order n, and the spread in the data points $\theta \equiv \xi_n - \xi_0$. Since lower order divided differences play no role in the series computation, the subject of propagating initial errors is not relevant. Instead, we examine the effects of round-off errors in each step of the series computation and obtain an overall error bound.

The algorithm involves many inner product computations. We consider two possible error conditions. In the first, the computed inner product $fl_2(\sum_{i=0}^{n} \alpha_i \beta_i)$ satisfies

$$\left| fl_2\left(\sum_{i=0}^{n} \alpha_i \beta_i\right) - \sum_{i=0}^{n} \alpha_i \beta_i \right| \leqslant \epsilon \sum_{i=0}^{n} |\alpha_i \beta_i| \tag{3.3.3}$$

for all n. The error analysis here is based upon methods presented by Wilkinson [1963] who takes ϵ as 1.06 times the machine precision. The error bound (3.3.3) holds, for example, when all additions are performed in double precision arithmetic (hence the subscript 2 in fl_2) and rounding to single precision is done only when the summation is completed.

In the second case, the computed sum $fl(\sum_{i=0}^{n} \alpha_i \beta_i)$ satisfies

$$\left| fl\left(\sum_{i=0}^{n} \alpha_i \beta_i\right) - \sum_{i=0}^{n} \alpha_i \beta_i \right| \leqslant \epsilon \sum_{i=0}^{n} (n+2-i) |\alpha_i \beta_i| . \tag{3.3.4}$$

This bound holds when the entire summation is performed in single precision arithmetic.

The following error bounds for $\Delta_0^n \exp_\tau$ are established in Appendix B. Under the double precision condition (3.3.3)

$$\left| fl_2(\Delta_0^n \exp_\tau) - \Delta_0^n \exp_\tau \right| \leqslant \epsilon (2 + \tau\theta/2) e^{\tau\theta/2} \frac{\tau^n e^{\tau\alpha}}{n!} . \tag{3.3.5}$$

and under the single precision condition (3.3.4)

$$\left| fl(\Delta_0^n \exp_\tau) - \Delta_0^n \exp_\tau \right| \leqslant \epsilon (m + n + 7 + \tau\theta/2) e^{\tau\theta/2} \frac{\tau^n e^{\tau\alpha}}{n!} . \tag{3.3.6}$$

The factor $\tau^n e^{\tau\alpha}/n!$ is $\Delta^n \exp_\tau$ evaluated for data points confluent at α. $m+1$ is the number of

terms added in our computed partial sum of the Taylor expansion and is chosen so that, say,

$$|\Delta_0^{n}\exp_\tau - \Delta_0^{n}s_{m+n}| = |e^{\tau\alpha}\sum_{j=m+1}^{\infty}\frac{\tau^{n+j}}{(n+j)!}\Delta_0^{n}\uparrow_\alpha^{n+j}| \leqslant \epsilon e^{\tau\theta/2}\frac{\tau^n e^{\tau\alpha}}{n!}.$$

m depends on ϵ, τ, and θ in a complicated way; only a general estimate can be given. For example Appendix B shows that when $\epsilon = 10^{-14}$ and $\tau\theta < 2$, $m \geqslant 16$ satisfies the above condition.

The bounds (3.3.5) and (3.3.6) are converted to relative error bounds by

$$\frac{\tau^n e^{\tau\xi_0}}{n!} \leqslant \Delta_0^{n}\exp_\tau \leqslant \frac{\tau^n e^{\tau\xi_n}}{n!},$$

which follows from $\Delta^{n}\exp_\tau$ being increasing in its abscissae. Then because $\xi_0 \leqslant \alpha \leqslant \xi_n$,

$$\frac{\tau^n e^{\tau\alpha}}{n!} \leqslant e^{\tau(\alpha-\xi_0)}\cdot\Delta_0^{n}\exp_\tau = e^{\tau\theta/2}\cdot\Delta_0^{n}\exp_\tau.$$

Relative error bounds for the Taylor series algorithm. The relative error ϵ_0^{n} in representing $\Delta_0^{n}\exp_\tau$ by its computed value satisfies

$$|\epsilon_0^{n}| \leqslant \epsilon(2+\tau\theta/2)e^{\tau\theta} \tag{3.3.7}$$

for double precision accumulation (3.3.3), and

$$|\epsilon_0^{n}| \leqslant \epsilon(m+n+7+\tau\theta/2)e^{\tau\theta} \tag{3.3.8}$$

for single precision accumulation (3.3.4).

The relative error bound in the first case does not depend on n. In both cases it is increasing in the "spread" $\tau\theta$. These error bounds are uniform in the sense that if the Taylor series algorithm were used to compute any other divided difference of the table (any $\Delta_i^{k}\exp_\tau$ for $k = 0, 1, \ldots, n$ and $i = 0, 1, \ldots, n-k$), a smaller error bound would result. This follows from the ordering condition $\xi_0 \leqslant \xi_i \leqslant \xi_n$. Error bounds for $\Delta_i^{k}\exp_\tau$ would either involve replacing n by $k \leqslant n$ or θ by a smaller number.

example: In Fig. 3.3.1, 8-th order divided differences correct to 7 digits are given initially for the standard scheme; the scheme is used only to compute the remaining higher order differences. The relative error increases by a factor of about 3 per step. Thus

$$|\epsilon_0^{n}| \approx 3^{n-8}\epsilon$$

n	abscissae ξ_n	Δ_0^nexp, correct to 7 decimal digits	Δ_0^nexp using standard scheme after $n=8$	Δ_0^nexp using Taylor series Algorithm	Δ_0^nexp using Algorithm of §3.4
0	-13.0	2.260329E-06			2.260332E-06
1	-12.5	2.932648E-06		2.917451E-06	2.932650E-06
2	-12.0	1.902471E-06		1.905751E-06	1.902472E-06
3	-11.5	8.227822E-07		8.215427E-07	8.227824E-07
4	-11.0	2.668782E-07		2.669856E-07	2.668782E-07
5	-10.5	6.925181E-08		6.925365E-08	6.925180E-08
6	-10.0	1.497504E-08		1.496362E-08	1.497505E-08
7	-9.5	2.775608E-09		2.777049E-09	2.775609E-09
8	-9.0	4.501490E-10	4.501490E-10	4.501040E-10	4.501491E-10
9	-8.5	6.489361E-11	6.489360E-11	6.490737E-11	6.489364E-11
10	-8.0	8.419572E-12	8.419580E-12	8.420343E-12	8.419573E-12
11	-7.5	9.930829E-13	9.930800E-13	9.930225E-13	9.930829E-13
12	-7.0	1.073723E-13	1.073727E-13	1.073735E-13	1.073724E-13
13	-6.5	1.071611E-14	1.071609E-14	1.071603E-14	1.071611E-14
14	-6.0	9.931098E-16	9.931271E-16	9.930869E-16	9.931100E-16
15	-5.5	8.590019E-17	8.590612E-17	8.590039E-17	8.590021E-17
16	-5.0	6.965660E-18	6.951898E-18	6.965612E-18	6.965660E-18
17	-4.5	5.316202E-19	5.402473E-19	5.316202E-19	5.316201E-19
18	-4.0	3.831926E-20	3.486964E-20	3.831920E-20	3.831926E-20
19	-3.5	2.616686E-21	3.650387E-21	2.616686E-21	2.616687E-21
20	-3.0	1.697500E-22	-6.986900E-23	1.697499E-22	1.697500E-22
21	-2.5	1.048766E-23	5.010054E-23	1.048766E-23	1.048766E-23
22	-2.0	6.185062E-25	-1.792933E-24	6.185061E-25	6.185063E-25
23	-1.5	3.489027E-26	-1.236419E-24	3.489027E-26	3.489027E-26
24	-1.0	1.886172E-27	6.496526E-25	1.886172E-27	1.886172E-27
25	-0.5	9.788799E-29	-1.931645E-25	9.788798E-29	9.788797E-29

Fig. 3.3.1: Top row of Δexp using several methods.

where $\epsilon \leqslant 5 \times 10^{-7}$, whereas the growth factor bound

$$\rho = 1 + \frac{2n}{\tau(\xi_n - \xi_0)}$$

discussed along with Fig. 3.2.3 gives a bound of 5 for the increase. The Taylor scheme yields good results for $n=25$ because the data points are symmetrically placed about the expansion point $\alpha=-7$; however, the lower order differences have less relative accuracy than Δ_δ^{25}exp. The relative error bound (3.3.7) with $\theta=12.5$ is

$$|\epsilon_0^n| \leqslant 2.3 \times 10^6 \epsilon .$$

For the lower order differences (small n), this overestimates $|\epsilon_0''|$ by a factor of about 10. Without a correct value of the divided difference to compare with, bounds such as the above must be accepted as the uncertainty in the computed divided differences, for all n.

The example shows that both the Taylor series algorithm and the standard scheme (even with some low order differences initially provided) may produce $\Delta_0''\exp_\tau$ with large relative errors when $\tau\theta$ is neither large nor small. The algorithm presented in the next section is designed to deal with this intermediate situation.

3.4 A scaling and squaring method.

At the end of the last section we saw that there are situations where neither the standard scheme nor the Taylor series algorithm yields a value of $\blacktriangle_0^n \exp_\tau$ with small relative error for all n of interest. We present here a third approach for computing divided differences of the exponential which, in many cases, gives significantly better error bounds. The method is based on the matrix function theorem for divided difference tables (§2.6) and is suggested in formula (2.6.8).

The entire divided difference table is representable as a function of the special "step matrix" Z (2.6.2). Specifically for $f = \exp_\tau$,

$$\blacktriangle \exp_\tau = \exp(\tau Z) \equiv e^{\tau Z} \tag{3.4.1}$$

where the diagonal of Z consists of the abscissae $\xi_0, \xi_1, \ldots, \xi_n$. Special properties of the exponential function are reflected in the divided difference table, denoted by $\blacktriangle \exp_\tau$. In particular for any non-negative integer j,

$$\blacktriangle \exp_\tau = [\exp(2^{-j}\tau Z)]^{2^j} = [\blacktriangle \exp_{2^{-j}\tau}]^{2^j}, \tag{3.4.2}$$

a formula for scaling and squaring the divided difference table. Ward [1977] has suggested scaling and squaring as a method for computing the exponential of a full matrix, whereas we propose to use it only in connection with Z.

example: With abscissae $\{0, 1, 2, 3, 4\}$,

$$\blacktriangle \exp_{1/2} = \begin{vmatrix} 1.0000 & 6.4872\text{E-}1 & 2.1042\text{E-}1 & 4.5501\text{E-}2 & 7.3794\text{E-}3 \\ & 1.6487 & 1.0696 & 3.4692\text{E-}1 & 7.5019\text{E-}2 \\ & & 2.7183 & 1.7634 & 5.7198\text{E-}1 \\ & & & 4.4817 & 2.9074 \\ & & & & 7.3891 \end{vmatrix}$$

to five digits. Squaring this matrix yields

$$(\blacktriangle \exp_{1/2})^2 = \begin{vmatrix} 1.0000 & 1.7183 & 1.4763 & 8.4553\text{E-}1 & 3.6322\text{E-}1 \\ & 2.7182 & 4.6709 & 4.0129 & 2.2984 \\ & & 7.3892 & 12.696 & 10.908 \\ & & & 20.086 & 34.513 \\ & & & & 54.599 \end{vmatrix}$$

For example,

$$\Delta_0^4\exp = \sum_{k=0}^{4} \Delta_0^k\exp_{1/2}\cdot\Delta_k^{4-k}\exp_{1/2}$$

$$= (1.0000)\times(0.0073794) + (0.64872)\times(0.075019) + (0.21042)\times(0.57198)$$

$$+ (0.045501)\times(2.9074) + (0.0073794)\times(7.3891)$$

$$= 0.36322;$$

and we check by (3.2.9)

$$\Delta_0^4\exp = \frac{1}{4!}(e^1-1)^4 = \frac{1}{4!}\times(1.7183)^4 = 0.36323.$$

Basic scaling and squaring algorithm. For (3.4.2) to be the basis of a useful algorithm, we first must obtain an initial divided difference table $\Delta\exp_{2^{-j}\tau}$. In the last section we saw that the relative error in the Taylor series algorithm decreases with the parameter τ. By choosing j large enough we can make, say, the error bound (3.3.7)

$$|\epsilon_0''| \leqslant \epsilon(2+2^{-(j+1)}\tau\theta)e^{2^{-j}\tau\theta}$$

small for any spread θ in the abscissae. Let β_j represent one of the coefficients of ϵ in (3.3.7) or (3.3.8), that is

$$\beta_j = (2+2^{-(j+1)}\tau\theta)e^{2^{-j}\tau\theta} \text{ or } \beta_j = (m+n+7+2^{-(j+1)}\tau\theta)e^{2^{-j}\tau\theta}. \tag{3.4.3}$$

The relative error bound $\beta_j\epsilon$ is uniform for every element of the divided difference table $\Delta\exp_{2^{-j}\tau}$ computed by the Taylor algorithm. Thus

$$|fl(\Delta\exp_{2^{-j}\tau}) - \Delta\exp_{2^{-j}\tau}| \leqslant \epsilon\beta_j\cdot\Delta\exp_{2^{-j}\tau}. \tag{3.4.4}$$

The inequality holds element by element.†

†For a matrix B, $|B|$ denotes the matrix all whose elements are the absolute values of the elements of B, i.e. $|B|_{i,j} = |B_{i,j}|$. Our notation $B \leqslant C$ means that $B_{i,j} \leqslant C_{i,j}$ for every i and j.

In §2.9 we remarked that it is not necessary to compute an entire divided difference table by the series algorithm. The Taylor algorithm need only produce the top row of $\Delta\exp_{2^{-1/\tau}}$. The backfill formula (2.9.1) generates the remainder of $\Delta\exp_{2^{-1/\tau}}$. Specifically when $\Delta_0^0\exp_{2^{-1/\tau}}, \ldots, \Delta_0^{n}\exp_{2^{-1/\tau}}$ are generated by the Taylor algorithm,

$$\Delta_i^k\exp_{2^{-1/\tau}} = (\xi_{i+k} - \xi_{i-1})\cdot\Delta_{i-1}^{k+1}\exp_{2^{-1/\tau}} + \Delta_{i-1}^{k}\exp_{2^{-1/\tau}} \tag{3.4.5}$$

for $k = 0, 1, \ldots, n-i$ are obtainable successively for $i = 1, 2, \ldots, n$. We show that this modification does not increase the error bound (3.4.4).

example:

$$\Delta\exp_{1/2} = \begin{vmatrix} 1.0000 & 6.4872\text{E-}1 & 2.1042\text{E-}1 & 4.5501\text{E-}2 & 7.3794\text{E-}3 \\ & 1.6487 & 1.0696 & 3.4692\text{E-}1 & 7.5019\text{E-}2 \\ & & 2.7183 & 1.7634 & 5.7198\text{E-}1 \\ & & & 4.4817 & 2.9074 \\ & & & & 7.3890 \end{vmatrix}$$

The top row is from the matrix in the previous example. The remainder of the table was filled in by (3.4.5). For example,

$$\begin{aligned} \Delta_1^3\exp_{1/2} &= (4-0)\cdot\Delta_0^4\exp_{1/2} + \Delta_0^3\exp_{1/2} \\ &= 4\times 0.0073794 + 0.045501 \\ &= 0.075019 \end{aligned}$$

Lemma: The relative error bound on every element of the divided difference table is no greater than the largest error bound on the elements in the top row of the table. This error bound is not increased when the table is filled out by the backfill scheme (3.4.5).

proof: Consider one step of the backfill routine

$$\Delta_1^{n-1}\exp_{2^{-1/\tau}} = (\xi_n - \xi_0)\cdot\Delta_0^n\exp_{2^{-1/\tau}} + \Delta_0^{n-1}\exp_{2^{-1/\tau}}.$$

Let ϵ_0'' be the relative error in the computed value of $\Delta_0''\exp_{2^{-l}\tau}$. The propagated error in $\Delta_1''^{-1}\exp_{2^{-l}\tau}$ is

$$\epsilon_1''^{-1}\cdot\Delta_1''^{-1}\exp_{2^{-l}\tau} = \epsilon_0''(\xi_n - \xi_0)\cdot\Delta_0''\exp_{2^{-l}\tau} + \epsilon_0''^{-1}\cdot\Delta_0''^{-1}\exp_{2^{-l}\tau}.$$

Both $|\epsilon_0''| \leqslant \epsilon\beta_j$ and $|\epsilon_0''^{-1}| \leqslant \epsilon\beta_j$, by the uniformity of the bound β_j for the Taylor series method. Thus

$$|\epsilon_1''^{-1}|\cdot\Delta_1''^{-1}\exp_{2^{-l}\tau} \leqslant \epsilon\beta_j\{(\xi_n - \xi_0)\cdot\Delta_0''\exp_{2^{-l}\tau} + \Delta_0''^{-1}\exp_{2^{-l}\tau}\},$$

so

$$|\epsilon_1''^{-1}| \leqslant \epsilon\beta_j.$$

When the abscissae are ordered $\xi_0 \leqslant \xi_1 \leqslant \cdots \leqslant \xi_n$, the sum in (3.4.5) involves non-negative numbers only and the above argument may be repeated. It shows that, considering only propagated errors, the uniform error bound (3.4.4) holds also when only the top row of $\Delta\exp_{2^{-l}\tau}$ is computed by the Taylor series algorithm and the remainder of the table backfilled according to (3.4.5). □

The outline of a new approach for computing the divided difference table $\Delta\exp_\tau$ is summarized as follows.

Algorithm 1: Scaling and squaring algorithm for $\Delta\exp_\tau$.

With the abscissae ordered such that $\xi_0 \leqslant \xi_1 \leqslant \cdots \leqslant \xi_n$,

1. Choose j and form $\Delta_0^k\exp_{2^{-l}\tau}$ for $k = 0, 1, \ldots, n$ by the Taylor series algorithm (§3.3);
2. Backfill the remainder of the table $\Delta\exp_{2^{-l}\tau}$ according to (3.4.5);
3. Square the divided difference table matrix j times.

Error bounds and selection of a scaling parameter. An error analysis of the algorithm shows how to select j. The elements of the divided difference table $\Delta\exp_\tau$ are non-negative for all $\tau \geqslant 0$. When inner products involved in matrix squaring are accumulated according to the error condition (3.3.3), not depending upon n, we have an element by element error bound

$$|fl_2(B^2) - B^2| \leqslant \epsilon B^2 \tag{3.4.6}$$

for any non-negative matrix B, such as $B = \Delta\exp_{2^{-j}}$. When a computed matrix $fl_2(B)$ satisfies

$$fl_2(B) - B \equiv E \text{ where } |E| \leqslant \epsilon\beta_1 B, \tag{3.4.7}$$

as in (3.4.4), then

$$|[fl_2(B)]^2 - B^2| = |BE + EB + E^2| \leqslant (2\epsilon\beta_1 + \epsilon^2\beta_1^2)B^2.$$

Thus squaring a computed matrix $fl_2(B)$ yields

$$\begin{aligned} |fl_2[fl_2(B)]^2 - B^2| &\leqslant |fl_2[fl_2(B)]^2 - [fl_2(B)]^2| + |[fl_2(B)]^2 - B^2| \\ &\leqslant \epsilon[fl_2(B)]^2 + (2\epsilon\beta_1 + \epsilon^2\beta_1^2)B^2 \\ &\leqslant \epsilon[(1+2\beta_1) + \epsilon(2+\beta_1)\beta_1 + \epsilon^2\beta_1^2]B^2. \end{aligned}$$

ϵ is so small that terms in ϵ^2 are negligible when compared with terms linear in ϵ. We take

$$|fl_2[fl_2(B)]^2 - B^2| \leqslant \epsilon(1+2\beta_1)B^2. \tag{3.4.8}$$

In (3.4.8) $B = \Delta\exp_{2^{-j}}$, so

$$fl_2[fl_2(B)]^2 = fl_2[fl_2(\Delta\exp_{2^{-j}})]^2 \equiv fl_2(\Delta\exp_{2^{-(j-1)}}).$$

The first computed matrix square satisfies

$$\begin{aligned} |fl_2(\Delta\exp_{2^{-(j-1)}}) - \Delta\exp_{2^{-(j-1)}}| &\leqslant \epsilon(1+2\beta_1)\cdot\Delta\exp_{2^{-(j-1)}} \\ &= \epsilon[2(\beta_1+1)-1]\cdot\Delta\exp_{2^{-(j-1)}}. \end{aligned}$$

This inequality is the same as (3.4.7), but now with $B = \Delta\exp_{2^{-(j-1)}}$ and β replaced by $\beta_{-1} \equiv 2(\beta_1+1)-1$. Hence iteratively

$$|fl_2(\Delta\exp_{2^{-(j-2)}}) - \Delta\exp_{2^{-(j-2)}}| \leqslant \epsilon[4(\beta_1+1)-1]\cdot\Delta\exp_{2^{-(j-2)}},$$

and after j steps

$$|fl_2(\Delta\exp_1) - \Delta\exp_1| \leqslant \epsilon[2^j(\beta_1+1)-1]\cdot\Delta\exp_1. \tag{3.4.9}$$

This last inequality is a relative error bound on the divided difference table computed according to the scaling and squaring algorithm. The bound depends on the Taylor series bound through β and increases exponentially in j, the number of squaring operations performed.

It is clear both from the error bound (3.4.9) and the amount of work involved in squaring matrices that we wish to choose j as small as possible. However β_j increases as j decreases. We demonstrate here how to select j to minimize the bound in (3.4.9). With j so chosen, we obtain an expression of the form $\kappa\tau\theta$ for the relative error bound. The value of the constant κ depends on the specific assumptions made in bounding round-off errors in the Taylor algorithm and the matrix squaring.

We want to minimize the coefficient in (3.4.9); that is we want to choose j to minimize the expression

$$g(j) \equiv 2^j(\beta_j + 1) - 1 = 2^j[(2 + 2^{-(j+1)}\tau\theta)e^{2^{-j}\tau\theta} + 1] - 1 .$$

Define $\sigma \equiv 2^{-j}\tau\theta$. τ and θ are fixed here, so minimizing $g(j)$ in j is equivalent to minimizing

$$\hat{g}(\sigma) \equiv \frac{1}{\sigma}[(2 + \sigma/2)e^{\sigma} + 1]$$

in σ. The minimum is $\hat{g}(\sigma) \approx 7.7885$ which occurs for $\sigma \approx 0.9606$. $2^{-j}\tau\theta = 0.9606$ is probably not true for integer values of j. Nevertheless for integer j, the $\sigma = 2^{-j}\tau\theta$ yielding the smallest value of $\hat{g}(\sigma)$ must satisfy $\sigma_0 < 2^{-j}\tau\theta \leqslant 2\sigma_0$ where $\hat{g}(\sigma_0) = \hat{g}(2\sigma_0)$.

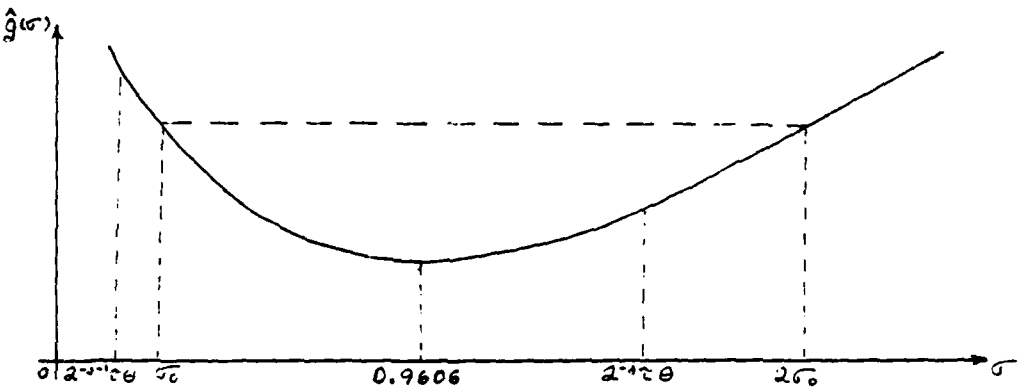


Fig. 3.4.1: Graph of $\hat{g}(\sigma)$ showing $(\sigma_0, 2\sigma_0]$ is the largest interval containing $\sigma = 2^{-j}\tau\theta$ for just one integer value of j.

$\sigma_0 \approx 0.6646$, so $0.6646 < 2^{-j}\tau\theta \leqslant 1.3292$. The minimizing j is the smallest non-negative integer satisfying

$$2^{-j}\tau\theta \leqslant 1.3292 . \tag{3.4.10}$$

For all $\sigma \in (0.6646, 1.3292]$, $\hat{g}(\sigma) \leqslant \hat{g}(1.3292)$. We are assured that for the above choice of j,

$\hat{g}(2^{-j}\tau\theta) \leqslant \hat{g}(1.3292) \approx 8.3259 \equiv \kappa_2$. Then

$$g(j) = \tau\theta \cdot \hat{g}(\sigma) - 1 \leqslant \kappa_2 \tau\theta - 1 .$$

Hence

$$|fl_2(\Delta\exp_\tau) - \Delta\exp_\tau| \leqslant \epsilon[\kappa_2\tau\theta - 1]\cdot\Delta\exp_\tau \tag{3.4.11}$$

when the scaling parameter j is chosen according to (3.4.10). This bound may fail when $\tau\theta$ is very small; in this case $j = 0$ and the Taylor series bound (3.3.7) is appropriate.

By a similar argument we derive a bound like (3.4.11) for the single precision error condition (3.3.4). In this case we write

$$\beta_j = (n+1)\gamma_j \equiv (m + n + 7 + 2^{-(j+1)}\tau\theta)e^{2^{-j}\tau\theta},$$

which is consistent with the other Taylor series error bound (3.3.8). The error in matrix squaring satisfies

$$|fl(B^2) - B^2| \leqslant \epsilon(n+1)B^2$$

for any non-negative matrix B. From (3.4.7),

$$fl(B) - B \equiv E \quad \text{where} \quad |E| \leqslant \epsilon(n+1)\gamma_j B .$$

Replacing ϵ by $\epsilon(n+1)$ and β_j by γ_j in our arguments leading up to (3.4.9),

$$|fl(\Delta\exp_\tau) - \Delta\exp_\tau| \leqslant \epsilon(n+1)[2^j(\gamma_j + 1) - 1]\cdot\Delta\exp_\tau . \tag{3.4.12}$$

In Appendix B we show that when $\epsilon \leqslant 10^{-7}$, m can be taken as small as 10. We assume also all first order divided differences are computed by a special formula as in (3.1.2), so our bounds here are applied only when $n \geqslant 2$. Also, j will be such that $2^{-(j+1)}\tau\theta < 1$. Hence

$$\gamma_j = (1 + \frac{m + 6 + 2^{-(j+1)}\tau\theta}{n+1})e^{2^{-j}\tau\theta} < 7e^{2^{-j}\tau\theta}. \tag{3.4.13}$$

As before we want an integer j to minimize the expression

$$2^j(7e^{2^{-j}\tau\theta} + 1) - 1 .$$

This is minimized when j is the smallest non-negative integer satisfying

$$2^{-j}\tau\theta \leqslant 1.4542. \tag{3.4.14}$$

For this j

$$|fl(\Delta\exp_\tau) - \Delta\exp_\tau| \leqslant \epsilon(n+1)[\kappa_1\tau\theta - 1]\cdot\Delta\exp_\tau \tag{3.4.15}$$

where $\kappa_1 = 21.2950$. The following box summarizes these bounds.

Scaling and squaring error bounds. For double precision accumulation of inner products (3.3.3),

$$|fl_2(\Delta\exp_\tau) - \Delta\exp_\tau| \leqslant \epsilon[\kappa_2\tau\theta - 1]\cdot\Delta\exp_\tau$$

where $\kappa_2 = 8.3259$, θ is the maximum spread in the abscissae, and the number of squarings j is the smallest non-negative integer such that

$$2^{-j}\tau\theta \leqslant 1.3292.$$

For single precision accumulation (3.3.4),

$$|fl(\Delta\exp_\tau) - \Delta\exp_\tau| \leqslant \epsilon(n+1)[\kappa_1\tau\theta - 1]\cdot\Delta\exp_\tau$$

where $\kappa_1 = 21.2950$ and j is the smallest non-negative integer such that

$$2^{-j}\tau\theta \leqslant 1.4542.$$

example: The entries in the right hand column of Fig. 3.3.1 were computed by scaling and squaring. The bound (3.4.11) has the coefficient

$$\kappa_2\tau\theta - 1 = 8.3259 \times 1 \times 12.5 - 1 \approx 103,$$

and $\log_{10}103 \approx 2.01$. This indicates a loss of two decimal digits, at most, in all the divided differences computed.

Modified scaling and squaring algorithm. The algorithm can be made more efficient by extending our use of the backfill scheme. Squaring a $(n+1)\times(n+1)$ triangular matrix (such as $\Delta\exp_{2^{-j}\tau}$) requires $(n+3)(n+2)(n+1)/6$ multiplications. The j squarings needed to get $\Delta\exp_\tau$ from $\Delta\exp_{2^{-j}\tau}$ involve $O(jn^3/6)$ operations. This operation count can be reduced to $O(jn^2)$.

Once the top (0-th) row of $fl_2(\Delta\exp_{2^{-(j-1)}\tau})$ is computed from squaring $fl_2(\Delta\exp_{2^{-j}\tau})$, the backfill scheme (2.9.1) will generate the remainder of $fl_2(\Delta\exp_{2^{-(j-1)}\tau})$ in exactly the same manner we generated the remainder of $fl_2(\Delta\exp_{2^{-j}\tau})$ given its top row by the Taylor series algorithm. Because relative errors in the elements in this top row of $fl_2(\Delta\exp_{2^{-(j-1)}\tau})$ are uniformly

bounded by $\epsilon(1+2\beta_j)$,

$$|fl_2(\Delta\exp_{2^{-(j-1)}\tau}) - \Delta\exp_{2^{-(j-1)}\tau}| \leqslant \epsilon(1+2\beta_j)\cdot\Delta\exp_{2^{-(j-1)}\tau}$$

for the entire table. This idea holds for all the squarings. We notice also that the uniform bounds are exactly those employed in the analysis of the scaling and squaring algorithm. Thus all our just derived error bounds are applicable when the matrix squaring is modified in the above manner. The same argument holds for single precision accumulation when ϵ is replaced by $\epsilon(n+1)$.

Now, how does the operation count change? Obtaining the top row of a matrix square requires $(n+2)(n+1)/2$ multiplications. Backfilling the rest of the matrix requires one multiplication per element, or $n(n+1)/2$ multiplications. Thus computing each matrix square by this modified method requires $(n+1)^2$ multiplications, compared with $(n+3)(n+2)(n+1)/6$ for the direct squaring approach. This is an improvement for all $n > 1$.

Algorithm 2: Modified scaling and squaring algorithm for $\Delta\exp_\tau$.

With the abscissae ordered such that $\xi_0 \leqslant \xi_1 \leqslant \cdots \leqslant \xi_n$,

1. Choose j according to (3.4.10) or (3.4.14) and form $\Delta_0^k\exp_{2^{-j}\tau}$ for $k = 0, 1, \ldots, n$ by the Taylor series algorithm (§3.3);
2. Backfill the remainder of the table $\Delta\exp_{2^{-j}\tau}$;
3. Square the divided difference table matrix $\Delta\exp_{2^{-j}\tau}$ j times in the modified manner: compute the 0-th row of the matrix square and then backfill the remainder of the table.

3.5 A hybrid algorithm for the divided difference table $\Delta\exp_\tau$.

We have now presented three quite different methods for computing $\Delta_0^n\exp_\tau \equiv (\Delta^n\exp_\tau)(\xi_0, \ldots, \xi_n)$: (1) standard divided differences, (2) Taylor series, and (3) scaling and squaring. These algorithms have complementary error propagation properties, but they vary in computational efficiency. We summarize here these two aspects of each algorithm and present a hybrid algorithm which may be used when none of the above alone is satisfactory for computing an entire table. For our hybrid algorithm we give error bounds which depend only on the order of the divided differences computed; these bounds are independent of the choice of abscissae and parameter τ.

(1) Standard. The propagated relative error ϵ_0^n in a typical step of the standard algorithm satisfies

$$|\epsilon_0^n| \leqslant |\epsilon_1^{n-1}| + r_0^n(\tau;x)\cdot[|\epsilon_1^{n-1}| + |\epsilon_0^{n-1}|],$$

where by (3.2.6)

$$r_0^n(\tau;x) \leqslant \frac{n}{\tau(\xi_n - \xi_0)}$$

when $\xi_n \geqslant \xi_i$ for all i.

When the abscissae are ordered, $\xi_0 \leqslant \xi_1 \leqslant \cdots \leqslant \xi_n$, and all initial relative errors ϵ_i^0 in the function values $\Delta_i^0\exp_\tau \equiv e^{\tau\epsilon_i}$ are uniformly bounded (that is $|\epsilon_i^0| \leqslant \epsilon$ for all i), we obtain a simple bound on ϵ_0^n. Let ϕ represent the minimum separation of the data points, that is $\phi \equiv \min_{1\leqslant i\leqslant n}(\xi_i - \xi_{i-1})$. The relative error in all first order divided differences satisfies

$$|\epsilon_i^1| \leqslant \epsilon(1 + 2/\tau\phi), \quad i = 0, 1, \ldots, n-1.$$

Continuing, we obtain

$$|\epsilon_0^n| \leqslant \epsilon\prod_{k=1}^{n}(1 + 2k/\tau\phi). \tag{3.5.1}$$

This bound is decreasing in $\tau\phi$. Given the (initial) $n+1$ exponentials $e^{\tau\epsilon_i}$, computing $\Delta_0^n\exp_\tau$, in fact the entire divided difference table, requires only $n(n+1)/2$ divisions.

(2) Series. The Taylor series method (§3.3) needs $2n+2$ multiplications (exclusive of coefficient evaluations) to add a new term to the partial sums we form for each $\Delta_0^k\exp_\tau$, $k = 0, 1, \ldots, n$. When each partial sum has $m+1$ terms, the total is $2m(n+1)$ multiplications

(the initial term requires no inner products) to get $\Delta_0^n \exp_\tau$, as well as all $\Delta_0^k \exp_\tau$ for $k = 0, 1, \ldots, n$.

(3) Scaling and squaring. Squaring a $(n+1) \times (n+1)$ divided difference table by the special method described in §3.4 requires $(n+1)^2$ multiplications, and we do this j times. For any but small n, the table squaring dominates the rest of the calculation.

Fig. 3.5.1 summarizes this information on bounds and operation counts. The error bounds listed in the second column assume that inner products satisfy the double precision error condition (3.3.3), that is

$$|fl_2(\sum_{i=0}^{n} \alpha_i \beta_i) - \sum_{i=0}^{n} \alpha_i \beta_i| \leqslant \epsilon \sum_{i=0}^{n} |\alpha_i \beta_i| .$$

Those in the third column reflect the single precision error condition (3.3.4), namely

$$|fl(\sum_{i=0}^{n} \alpha_i \beta_i) - \sum_{i=0}^{n} \alpha_i \beta_i| \leqslant \epsilon \sum_{i=0}^{n} (n+2-i) |\alpha_i \beta_i| .$$

The bounds depend on the minimum separation ϕ and the maximum spread θ in the abscissae. The constants κ_2 and κ_1 in the bottom entries depend on the details of the arithmetic in the scaling and squaring algorithm. In §3.4 we derived the values $\kappa_2 = 8.3259$, and $\kappa_1 = 21.2950$ when $\epsilon = 10^{-7}$.

Method	Relative error bound coefficients		Operations
	Double precision	Single precision	
Standard algorithm with minimum separation ϕ	$\prod_{k=1}^{n} (1 + 2k/\tau\phi)$	$\prod_{k=1}^{n} (1 + 2k/\tau\phi)$	$\sim \frac{1}{2} n^2$
Taylor series, using $m+1$ terms, with spread θ	$(2 + \tau\theta/2) e^{\tau\theta}$	$(m + n + 7 + \tau\theta/2) e^{\tau\theta}$	$\sim 2mn$
Scaling and squaring with spread θ	$\kappa_2 \tau\theta - 1$	$(n+1)[\kappa_1 \tau\theta - 1]$	$\sim jn^2$

Fig. 3.5.1: Summary of three divided difference algorithms for $\Delta_0^n \exp_\tau$.

Decision criteria and the hybrid algorithm. Our error bounds suggest a hybrid algorithm: compute all *divided differences* having closely clustered abscissae by scaling and squaring, and the

remainder of the table by the standard formula. The Taylor series is a special case of scaling and squaring with $j=0$. The operation counts suggest employing scaling and squaring on divided differences of the smallest practical order. Our desire for good accuracy and our desire for efficiency, however, are in conflict. Here we lean towards the former in presenting criteria for deciding which method to employ when computing a particular difference in the table; we develop an overall error bound which holds for any sequence of abscissae and parameter τ.

A simple criterion is to use the "best" method to compute each divided difference in the table. By best we mean that method (either scaling and squaring or standard) which yields the smallest relative error bound for that particular divided difference we are considering. All lower order divided differences are assumed to have been computed already by the best method, or by a special formula.

For example in computing $\Delta_0^j \exp_\tau$ with double precision accumulation, we use scaling and squaring when

$$(\kappa_2 \tau\theta - 1)\epsilon < (1 + 2/\tau\theta)\epsilon ,$$

and the standard algorithm, otherwise. Here $\theta = \xi_1 - \xi_0$ is both the spread and the minimum separation in the data points. The worst possible error bound for this hybrid, then, occurs when

$$\kappa_2 \tau\theta - 1 = 1 + 2/\tau\theta .$$

This has the solution

$$\tau\theta = (1 + \sqrt{1 + 2\kappa_2})/\kappa_2 \equiv \tau\theta_1 .$$

For $\kappa_2 = 8.3259$ (as derived in §3.4), $\tau\theta_1 \approx 0.62$ and

$$\kappa_2 \tau\theta_1 - 1 \approx 1 + 2/\tau\theta_1 \approx 4.20 .$$

Thus the relative error ϵ_0^j is bounded, $|\epsilon_0^j| \leqslant 4.20\epsilon$, when the "best" method is used. This bound does not depend on the abscissae or τ, only on the value of κ_2. We have obtained a bound independent of the abscissae and τ when we use scaling and squaring for $\tau\theta < \tau\theta_1$ and the standard formula for $\tau\theta \geqslant \tau\theta_1$. This is a simple criterion for deciding which method to employ.

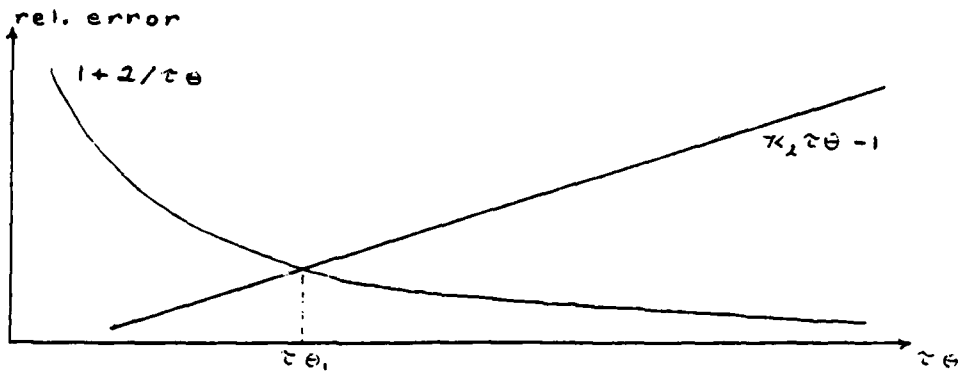


Fig. 3.5.2: Uncertainty in computed values of $\Delta_0^1 \exp_\tau$.

It is important to note exactly what our criterion means. The case $\tau\theta = \tau\theta_1$ does not mean that the two methods are equally accurate, only that our convenient error bounds for each method are equal. Each bound may be viewed as our maximum uncertainty in the computed $\Delta_0^1 \exp_\tau$ when the appropriate method is used. Thus when $\tau\theta = \tau\theta_1$ our uncertainty is equalized for the two methods, and is maximized over all $\tau\theta$ for the hybrid method. The number 4.20ϵ, for example, represents our maximum uncertainty in the computed $\Delta_0^1 \exp_\tau$ when the "best" method is used. More refined error bounds using information about τ and the abscissae will reduce the uncertainty, but at the loss of the simplicity we have here.

For $\Delta_0^2 \exp_\tau$ and $\theta = \xi_2 - \xi_0$, the relative error for the standard formula is bounded by $(\kappa_2\tau\theta_1 - 1)(1 + 4/\tau\theta)\epsilon$. We use scaling and squaring when

$$(\kappa_2\tau\theta - 1)\epsilon < (\kappa_2\tau\theta_1 - 1)(1 + 4/\tau\theta)\epsilon .$$

The largest error bound occurs when equality holds. Let this happen for $\tau\theta = \tau\theta_2$, thus

$$\kappa_2\tau\theta_2 - 1 = (\kappa_2\tau\theta_1 - 1)(1 + 4/\tau\theta_2) .$$

This procedure may be followed for all divided differences. For $\Delta_0^n \exp_\tau$ we obtain the recurrence in $\tau\theta_n$,

$$\kappa_2\tau\theta_n - 1 = (\kappa_2\tau\theta_{n-1} - 1)(1 + 2n/\tau\theta_n) . \tag{3.5.2}$$

The criterion for scaling and squaring in computing $\Delta_0^n \exp_\tau$ is

$$\tau(\xi_n - \xi_0) < \tau\theta_n , \tag{3.5.3}$$

n	$\tau\theta_n$	Error bound coefficient	Bound on decimal digits lost	$n(n-3)$	$\log_{10}[\kappa_2 n(n-3)-1]$
1	0.62	4.20	0.62	-2	
2	1.77	13.71	1.14	-2	
3	4.15	33.54	1.53	0	
4	8.12	66.59	1.82	4	1.51
5	13.88	114.57	2.06	10	1.92
6	21.55	178.38	2.25	18	2.17
7	31.17	258.51	2.41	28	2.37
8	42.78	355.19	2.55	40	2.52
9	56.40	468.55	2.67	54	2.65
10	72.02	598.66	2.78	70	2.76
11	89.67	745.55	2.87	88	2.86
12	109.32	909.22	2.96	108	2.95
13	131.00	1,089.68	3.04	130	3.03
14	154.69	1,286.92	3.11	154	3.11
15	180.39	1,500.94	3.18	180	3.18
16	208.11	1,731.73	3.24	208	3.24
17	237.85	1,979.28	3.30	238	3.30
18	269.59	2,243.58	3.35	270	3.35
19	303.35	2,524.63	3.40	304	3.40
20	339.11	2,822.43	3.45	340	3.45
21	376.89	3,136.95	3.50	378	3.50
22	416.68	3,468.21	3.54	418	3.54
23	458.47	3,816.18	3.58	460	3.58
24	502.27	4,180.88	3.62	504	3.62
25	548.08	4,562.29	3.66	550	3.66
30	807.23	6,719.91	3.83	810	3.83
35	1,116.50	9,294.88	3.97	1,120	3.97
40	1,475.87	12,286.96	4.09	1,480	4.09
45	1,885.31	15,695.94	4.20	1,890	4.20
50	2,344.82	19,521.72	4.29	2,350	4.29
60	3,413.96	28,423.30	4.45	3,420	4.45
70	4,683.24	38,991.19	4.59	4,690	4.59
80	6,152.62	51,225.09	4.71	6,160	4.71
90	7,822.07	65,124.80	4.81	7,830	4.81
100	9,691.59	80,690.18	4.91	9,700	4.91

Fig. 3.5.3: Error bounds and decision criteria for hybrid algorithm.

and the relative error is bounded by $(\kappa_2\tau\theta_n - 1)\epsilon$. Since $\kappa_2\tau\theta_0 - 1 \equiv 1$, the recurrence (3.5.2) can be evaluated to yield $\tau\theta_n$ for any n. These values, along with corresponding relative error bounds, are listed in Fig. 3.5.3 when $\kappa_2 = 8.3259$.

We can now summarize a complete algorithm for computing divided differences of the exponential with real abscissae.

Algorithm: Hybrid algorithm for $\Delta\exp_\tau$.

1. Compute $\Delta_i^0 = e^{\tau\xi_i}$ for each $i = 0, 1, \ldots, n$.
2. For $k = 1, 2, \ldots, n$ and $i = 0, 1, \ldots, n-k$, when
$$\tau(\xi_{i+k} - \xi_i) < \tau\theta_k$$
compute $\Delta_i^k\exp_\tau$ by scaling and squaring; otherwise, when
$$\tau(\xi_{i+k} - \xi_i) \geqslant \tau\theta_k$$
compute $\Delta_i^k\exp_\tau$ by the standard formula.

The hybrid algorithm requires us to decide which divided difference scheme to use for each divided difference. For example in computing $\Delta_0^{12}\exp_\tau$ by employing the values in Fig. 3.5.3, when all lower order divided differences have been computed according to the algorithm, scaling and squaring is used when

$$\tau(\xi_{12} - \xi_0) < \tau\theta_{12} \approx 109.32.$$

The standard scheme is used otherwise. The relative error in our computed $\Delta_0^{12}\exp_\tau$, that is ϵ_0^{12}, satisfies

$$|\epsilon_0^{12}| \leqslant (\kappa_2\tau\theta_{12} - 1)\epsilon \approx 909.22\epsilon.$$

$\log_{10}(909.22) \approx 3$ bounds the number of decimal digits lost in computing $\Delta_0^{12}\exp_\tau$ by the hybrid algorithm. That is, when all $\Delta^0\exp_\tau$ are given to 10 correct decimal digits, say, our computed $\Delta_0^{12}\exp_\tau$ contains, **at least**, 7 correct decimal digits.

To gain a better idea of how the decision criterion $\tau\theta_n$, and its associated error bound $\kappa_2\tau\theta_n - 1$, depend upon the order of the divided difference n, we bound solutions of the recurrence (3.5.2). Appendix C shows that

$$\tau\theta_n < n^2 + n + \frac{2}{\kappa_2} \tag{3.5.4}$$

for $n \geqslant 1$ and any $\kappa_2 > 0$. Hence the relative error in $\Delta_0^n\exp_\tau$, computed according to the hybrid algorithm with double precision accumulation, satisfies

$$|\epsilon_0^n| < [\kappa_2(n^2 + n) + 1]\epsilon. \tag{3.5.5}$$

The relative error, then, increases in n, at worst, as $O(n^2)$. This bound holds regardless of our

choice of abscissae and parameter τ. For our example with $\kappa_2 = 8.3259$,

$$n(n-4) < \tau\theta_n < n(n-3)$$

when $n \geqslant 17$ (Appendix C). For comparison, the last two columns of Fig. 3.5.3 contain values of $n(n-3)$ and $\log_{10}[\kappa_2 n(n-3)-1]$. The latter numbers closely bound the digits lost values for large n.

Single precision decision criteria. A similar analysis in the case of single precision accumulation shows that the hybrid algorithm error bound behaves as $O(n^3)$. Here the scaling and squaring error satisfies

$$|\epsilon_0''| \leqslant (n+1)(\kappa_1\tau\theta - 1)\epsilon\,,$$

but the bound on the standard scheme is unchanged. The same argument as before leads, now, to the recurrence

$$(n+1)(\kappa_1\tau\theta_n - 1) = n(\kappa_1\tau\theta_{n-1} - 1)(1 + 2n/\tau\theta_n)\,. \tag{3.5.6}$$

In deriving the scaling and squaring bound we assumed all first order differences are computed by a special method, therefore we require initially

$$2(\kappa_1\tau\theta_1 - 1) = 1\,;$$

hence

$$\tau\theta_1 = \frac{3}{2\kappa_1}$$

is the initial value. The table in Fig.3.5.4 lists values of the decision criterion $\tau\theta_n$ and its associated error bound for $\kappa_1 = 21.2950$, which was derived in §3.4.

We can also show how $\tau\theta_n$ and its associated error bound for the hybrid algorithm depend upon n in the single precision accumulation case. From Appendix C,

$$\tau\theta_n \leqslant \frac{2}{3}n^2 + (\frac{3}{2\kappa_1} - \frac{2}{3})n \tag{3.5.7}$$

for $n \geqslant 1$ and all $\kappa_1 > 0$; hence

$$|\epsilon_0''| \leqslant \epsilon(n+1)[\frac{2}{3}\kappa_1 n^2 + (\frac{3}{2} - \frac{2\kappa_1}{3})n - 1] \tag{3.5.8}$$

n	$\tau\theta_n$	Error bound coefficient	Bound on decimal digits lost	$n(2n-5)/3$	$\log_{10}[(n+1)\{\kappa_1 \cdot n(2n-5)/3-1\}]$
1	0.07	1.00	0.00	-1.00	
2	0.28	15.11	1.18	-0.67	
3	1.15	93.95	1.97	1.00	1.91
4	3.16	331.66	2.52	4.00	2.62
5	6.58	835.34	2.92	8.33	3.02
6	11.50	1,707.08	3.23	14.00	3.32
7	17.90	3,041.99	3.48	21.00	3.55
8	25.77	4,930.48	3.69	29.00	3.75
9	35.08	7,460.36	3.87	39.00	3.92
10	45.80	10,717.98	4.03	50.00	4.07
11	57.92	14,789.00	4.17	62.33	4.20
12	71.42	19,758.68	4.30	76.00	4.32
13	86.29	25,712.06	4.41	91.00	4.43
14	102.53	32,734.11	4.52	107.33	4.53
15	120.12	40,909.77	4.61	125.00	4.63
16	139.06	50,323.94	4.70	144.00	4.72
17	159.35	61,061.56	4.79	164.33	4.80
18	180.98	73,207.55	4.86	186.00	4.88
19	203.96	86,846.88	4.94	209.00	4.95
20	228.28	102,064.51	5.01	233.33	5.02
21	253.94	118,945.43	5.08	259.00	5.08
22	280.93	137,574.66	5.14	286.00	5.15
23	309.27	158,037.22	5.20	314.33	5.21
24	338.94	180,418.14	5.26	344.00	5.26
25	369.95	204,802.47	5.31	375.00	5.32
30	545.01	359,752.88	5.56	550.00	5.56
35	753.42	577,551.81	5.76	758.33	5.76
40	995.17	868,839.39	5.94	1,000.00	5.94
45	1,270.26	1,244,258.17	6.09	1,275.00	6.10
50	1,578.67	1,714,452.19	6.23	1,583.33	6.24
60	2,295.47	2,981,746.70	6.47	2,300.00	6.48
70	3,145.59	4,755,889.66	6.68	3,150.00	6.68
80	4,129.02	7,122,053.25	6.85	4,133.33	6.85
90	5,245.78	10,165,412.46	7.01	5,250.00	7.01
100	6,495.85	13,971,143.98	7.15	6,500.00	7.15

Fig. 3.5.4: Single precision decision criteria and error bounds for the hybrid algorithm.

for the hybrid algorithm. Thus the relative error is, at worst, $O(n^3)$ in n. We stress that this bound holds for any choice of abscissae and parameter τ. Further, in the example with $\kappa_1 = 21.2950$, Appendix C shows that

$$\frac{2}{3}n^2 - 2n < \tau\theta_n < \frac{2}{3}n^2 - \frac{5}{3}n$$

for $n \geqslant 15$. The rightmost two columns of Fig. 3.5.4 list values of $n(2n-5)/3$ and $\log_{10}[(n+1)\{\kappa_1 n(2n-5)/3-1\}]$.

The hybrid algorithm demonstrates that it is possible to compute exponential divided differences to a desired accuracy. Our error bounds, particularly the digits lost bounds, tell how many decimal digits we must carry in order to be assured that $\Delta_0''\exp_\tau$ has desired accuracy. A short discussion of some useful modifications of the basic hybrid algorithm follows in the next section, along with a numerical example in which a rather large divided difference table is computed.

3.6 Comments on the hybrid divided difference algorithm.

The hybrid algorithm of the last section demonstrates that we can compute high order exponential divided differences with only a modest loss of precision. For this reason, the algorithm is a valuable theoretical tool. In this section we propose some modifications which make its implementation more efficient.

We gave scant consideration to computational efficiency when deriving the hybrid algorithm. Our error bounds and decision criteria apply without reference to any particular sequence of data points or parameter τ. As a result, the algorithm recomputes low order divided differences when scaling and squaring is used for differences whose "patterns of dependence" overlap. Also, the decision criteria are based upon worst case arrangements of the abscissae. These arrangements cannot be achieved since it is impossible to arrange even three points on a line such that their separations are quadratic. A relaxed decision criterion may greatly increase efficiency, without sacrificing accuracy. We now propose a possible modification to the algorithm by introducing an arbitrary criterion to cluster the data points.

Clustering. Let g be a positive increasing function of the order k of the divided difference under consideration. We decide to use scaling and squaring to compute $\Delta_i^k \exp_\tau$ when

$$\tau(\xi_{i+k} - \xi_i) < g(k); \tag{3.6.1}$$

otherwise, we use the standard formula. In addition, however, we do not permit the computation of overlapping table blocks by scaling and squaring. For example, suppose the decision criterion (3.6.1) demands that both $\Delta_i^k \exp_\tau$ and $\Delta_j^l \exp_\tau$, with $i \leqslant j \leqslant i+k \leqslant j+l$, be computed by scaling and squaring. We compute only $\Delta_i^{j+l-i} \exp_\tau$ by scaling and squaring, regardless of whether or not

$$\tau(\xi_{j+l} - \xi_i) < g(j+l-i).$$

The picture in Fig. 3.6.1 shows how overlapping blocks may be combined. We now speak of the abscissae $\xi_i, \xi_{i+1}, \ldots, \xi_{j+l}$ as being "clustered," and refer to the block of the divided difference table formed by $\Delta_i^{j+l-i} \exp_\tau$'s pattern of dependence as corresponding to this cluster of abscissae. So when two clusters overlap, they are combined.

When the clustering procedure is completed, the resulting clusters have no abscissae in common; the corresponding blocks in the table do not overlap. This clustering depends on the abscissae, not on the divided differences, and it may be performed prior to any divided

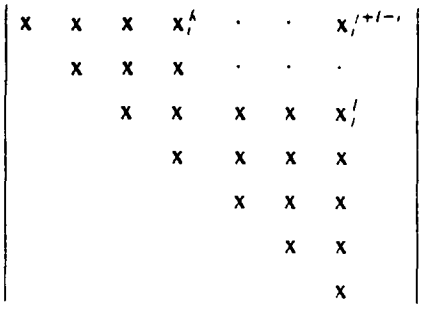


Fig. 3.6.1: Clustering of overlapping blocks in a table.

difference computations. Table blocks corresponding to the resulting clusters are computed by scaling and squaring with backfill. The picture in Fig. 3.6.2 shows what the table might then resemble. The remainder of the table is filled in by the standard scheme.

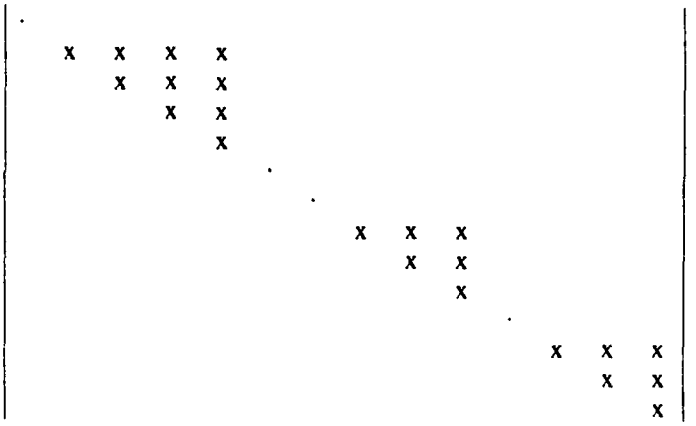


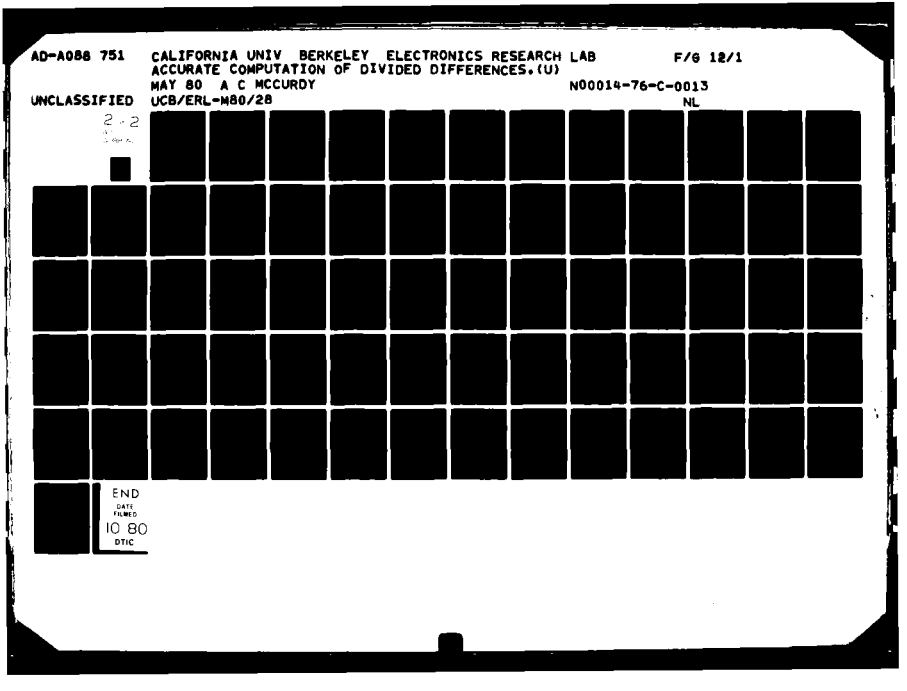
Fig. 3.6.2: Block structure of a divided difference table after the scaling and squaring step.

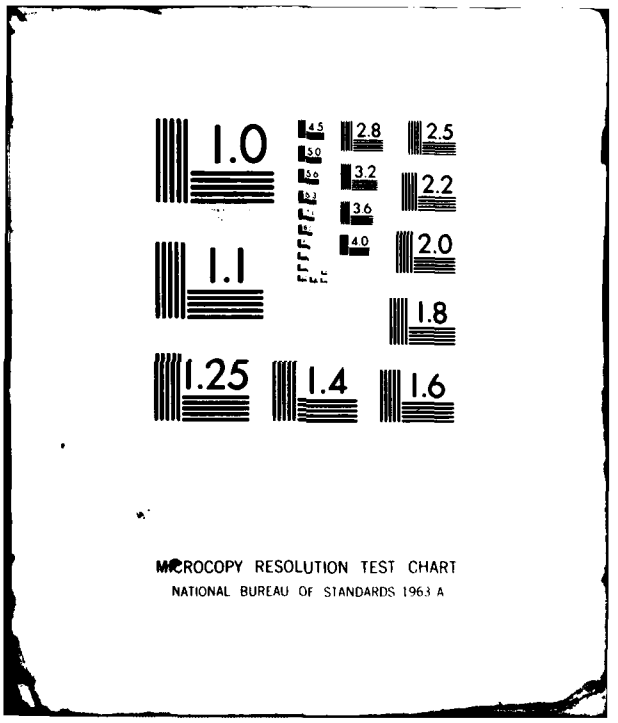
Our error bounds make possible a quick a priori error bound computation. For example, the bound (3.4.11) may be used when scaling and squaring is indicated, and the iterative bound (3.2.10) when the standard scheme is called for. We may even wish to compute the differences using the decision criterion (3.6.1) and then examine a postiori error bounds. In any event, when these bounds are unacceptable the original hybrid algorithm does guarantee a bound on the error and may be used when more efficient alternatives fail.

Divided Differences by the Hybrid Algorithm						
n	Abscissae ξ_n	Clustering	$\Delta_0^n \exp$ by algorithm	Correct values to 7 digits	A priori bound	Relative error
0	-34.5	0	1.039538E-15	1.039538E-15	0.00	0.00
1	-33.1	0	2.268571E-15	2.268571E-15	0.50	0.00
2	32.9	0	1.498804E-15	1.498804E-15	2.00	0.00
3	-14.4	3	8.015853E-11	8.015853E-11	0.77	0.00
4	-14.4	3	6.755117E-11	6.755117E-11	0.69	0.00
5	-14.4	3	2.879424E-11	2.879424E-11	1.61	0.21
6	-14.4	3	8.262803E-12	8.262803E-12	1.83	0.00
7	-14.1	3	1.891783E-12	1.891783E-12	1.96	0.06
8	6.1	8	2.013388E-09	2.013388E-09	1.23	0.00
9	6.4	8	1.522937E-09	1.522937E-09	1.05	0.32
10	6.8	8	6.118262E-10	6.118264E-09	3.54	0.78
11	7.1	8	1.663523E-10	1.663523E-09	3.95	0.94
12	11.3	8	7.590633E-11	7.590636E-11	4.01	0.92
13	11.3	8	1.883713E-11	1.883713E-11	4.16	0.58
14	11.3	8	3.359880E-12	3.359880E-12	4.35	0.15
15	12.2	8	5.323018E-13	5.323021E-13	4.51	0.90
16	12.2	8	6.841381E-14	6.841383E-14	4.70	0.65
17	13.1	8	8.156695E-15	8.156692E-15	4.85	0.89
18	25.6	18	5.861817E-15	5.861819E-15	4.61	0.50
19	28.7	19	2.750415E-15	2.750417E-15	4.42	1.16
20	32.9	20	1.381999E-15	1.382000E-15	4.22	0.73
21	33.4	20	3.448419E-16	3.448422E-16	4.07	1.17
22	33.4	20	5.740436E-17	5.740418E-17	4.21	1.71
23	34.5	20	8.338935E-18	8.339004E-18	4.72	2.14

Fig. 3.6.3: Example of the hybrid algorithm with clustering for $\tau = 1$.

example: The modified hybrid algorithm, with clustering, is illustrated in Fig. 3.6.3 for a collection of abscissae which includes confluent, close and well-separated data points. The clustering function is $g(k) \equiv k$. The third column of Fig. 3.6.3 indicates the resulting clustering of the abscissae. The fourth column contains the top row of the divided difference table computed in single precision with about seven decimal digits. The fifth has, for comparison, the same differences computed in double precision. Finally, a priori error bounds, calculated from (3.4.15) and a growth factor bound from (3.2.8), and the actual relative error are given in a digits lost form. Complete tables corresponding to Fig. 3.6.3 are presented in Appendix D. Fig. 3.6.4 repeats the same computation, but with $\tau = 2$. Finally, Fig. 3.6.5 shows the result of computing the entire table in one scaling and squaring for $\Delta_0^{23}\exp_2$.


AD-A088 751
CALIFORNIA UNIV BERKELEY ELECTRONICS RESEARCH LAB F/G 12/1
ACCURATE COMPUTATION OF DIVIDED DIFFERENCES.(U)
MAY 80 A C MCCURDY
N00014-76-C-0013
UNCLASSIFIED
UCB/ERL-M80/28
NL
2 - 2
END
DATE FILMED
10 80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Divided Differences by the Hybrid Algorithm						
n	Abscissae ξ_n	Clustering	$\Delta_0^n \exp_2$ by algorithm	Correct values to 7 digits	A priori bound	Relative error
0	-34.5	0	1.080639E-30	1.080639E-30	0.00	0.00
1	-33.1	1	1.192152E-29	1.192152E-29	0.50	0.00
2	-32.9	1	1.986183E-29	1.986183E-29	0.77	0.26
3	-14.4	3	4.467966E-17	4.467966E-17	0.51	0.16
4	-14.4	3	8.233205E-17	8.233205E-17	0.58	0.00
5	-14.4	3	7.604186E-17	7.604185E-17	1.85	0.13
6	-14.4	3	4.692805E-17	4.692805E-17	1.97	0.08
7	-14.1	3	2.444065E-17	2.444065E-17	2.03	0.07
8	6.1	8	8.977370E-07	8.977370E-07	0.95	0.00
9	6.4	8	1.963505E-06	1.963505E-06	0.72	0.00
10	6.8	8	2.309258E-06	2.309258E-06	2.38	0.50
11	7.1	8	1.760567E-06	1.760566E-06	2.59	0.32
12	11.3	12	1.305495E-05	1.305495E-05	2.23	0.00
13	11.3	12	1.217449E-05	1.217449E-05	1.99	0.53
14	11.3	12	6.558480E-06	6.558482E-06	3.41	0.80
15	12.2	12	3.453429E-06	3.453430E-06	3.86	0.43
16	12.2	12	1.281568E-06	1.281569E-06	4.26	0.98
17	13.1	12	4.751205E-07	4.751204E-07	4.48	0.55
18	25.6	18	9.602053E-04	9.602055E-04	3.93	0.63
19	28.7	19	1.412638E-02	1.412640E-02	3.28	1.17
20	32.9	20	4.335487E-01	4.335489E-01	2.54	0.93
21	33.4	20	6.106924E-01	6.106929E-01	1.92	1.04
22	33.4	20	3.836830E-01	3.836832E-01	2.55	0.84
23	34.5	24	2.381056E-01	2.381055E-01	2.94	0.54

Fig. 3.6.4: *Example of the hybrid algorithm with clustering for* $\tau = 2$.

Special methods for low order differences. It is sometimes possible to compute low order divided differences by a special formula. From (3.1.2) where

$$\Delta_0^1 \exp_\tau = e^{\tau(\xi_1+\xi_0)/2} \frac{\sinh[\tau(\xi_1-\xi_0)/2]}{(\xi_1-\xi_0)/2},$$

we see that first order differences may always be computed accurately when a good sinh function is available. Error growth in using the standard divided difference formula is primarily dependent on errors propagated from low order differences. Special computation of these differences may be very effective† in reducing errors in higher differences and in extending the

†Fig. 3.8.3 gives an example (for complex abscissae) of dramatic improvement in the error when first order divided differences are computed by a special formula.

area over which this simple formula may be used. In addition, scaling and squaring never need be used for these low order differences.

Divided Differences by Scaling and Squaring						
n	Abscissae ξ_n	Clustering	$\Delta_0^n \exp_2$ by algorithm	Correct values to 7 digits	A priori bound	Relative error
0	-34.5	0	1.080639E-30	1.080639E-30	0.00	0.00
1	-33.1	0	1.192152E-29	1.192152E-29	0.50	0.00
2	-32.9	0	1.986174E-29	1.986183E-29	4.85	1.88
3	-14.4	0	4.467955E-17	4.467966E-17	4.85	1.60
4	-14.4	0	8.233186E-17	8.233205E-17	4.85	1.57
5	-14.4	0	7.604169E-17	7.604185E-17	4.85	1.55
6	-14.4	0	4.692796E-17	4.692805E-17	4.85	1.50
7	-14.1	0	2.444060E-17	2.444065E-17	4.85	1.56
8	6.1	0	8.977336E-07	8.977370E-07	4.85	1.80
9	6.4	0	1.963496E-06	1.963505E-06	4.85	1.89
10	6.8	0	2.309243E-06	2.309258E-06	4.85	2.06
11	7.1	0	1.760551E-06	1.760566E-06	4.85	2.16
12	11.3	0	1.305484E-05	1.305495E-05	4.85	2.13
13	11.3	0	1.217439E-05	1.217449E-05	4.85	2.12
14	11.3	0	6.558430E-06	6.558482E-06	4.85	2.12
15	12.2	0	3.453403E-06	3.453430E-06	4.85	2.11
16	12.2	0	1.281559E-06	1.281569E-06	4.85	2.11
17	13.1	0	4.751175E-07	4.751204E-07	4.85	2.02
18	25.6	0	9.601986E-04	9.602055E-04	4.85	2.08
19	28.7	0	1.412635E-02	1.412640E-02	4.85	1.71
20	32.9	0	4.335491E-01	4.335489E-01	4.85	1.00
21	33.4	0	6.106924E-01	6.106929E-01	4.85	1.04
22	33.4	0	3.836829E-01	3.836832E-01	4.85	1.13
23	34.5	0	2.381053E-01	2.381055E-01	4.85	1.09

Fig. 3.6.5: Example of the scaling and squaring algorithm for $\tau = 2$.

Second order differences also may be computed accurately by a special formula when a routine is available to evaluate the function

$$h(\xi) \equiv \frac{e^{\xi}-1}{\xi} - 1 = \sum_{j=1}^{\infty} \frac{\xi^j}{(j+1)!}$$

accurately for all ξ. Let $\xi_0 \leqslant \xi_1 \leqslant \xi_2$, then

$$\Delta_0^2 \exp_\tau = e^{\tau\xi_1}\left\{\frac{e^{\tau(\xi_2-\xi_1)}-1}{\xi_2-\xi_1} - \frac{1-e^{\tau(\xi_0-\xi_1)}}{\xi_1-\xi_0}\right\}$$

$$= \tau e^{\tau\xi_1}\{(\frac{e^{\tau(\xi_2-\xi_1)}-1}{\tau(\xi_2-\xi_1)} - 1) - (\frac{e^{\tau(\xi_0-\xi_1)}-1}{\tau(\xi_0-\xi_1)} - 1)\}$$

$$= \tau e^{\tau\xi_1}\{h[\tau(\xi_2-\xi_1)] - h[\tau(\xi_0-\xi_1)]\}.$$

Because $\text{sign}[h(\xi)] = \text{sign}(\xi)$ and $\xi_2-\xi_1 \geq 0$ while $\xi_0-\xi_1 \leq 0$, the subtraction is actually an addition of non-negative numbers.

Perturbations and shifts in the abscissae. Abscissae used in computing divided differences may be obtained either experimentally, or as the result of earlier computations. In either case we may be uncertain what are the exact abscissae (represented here by the vector $\hat{x}$, say). The abscissae, say x, we have in hand are only approximations. The most we can expect is to have a bound in terms of x on our uncertainty in the value of $\hat{x}$. Thus given x and a bound on the uncertainty, we ask how far can the divided difference $\Delta^n\exp_\tau(x)$ be from $\Delta^n\exp_\tau(\hat{x})$. That is, how unsure are we of the value of a divided difference, given our doubt about its data.

As an example, we have presented without comment several formulas in which abscissae are shifted by a constant amount, say α. In finite precision arithmetic, a computed shifted abscissa $fl(\xi+\alpha)$ satisfies

$$|fl(\xi+\alpha) - (\xi+\alpha)| \leq (|\xi|+|\alpha|)\epsilon.$$

To have a uniform bound for all abscissae represented in a vector x, we write

$$\|fl(x+\alpha u) - (x+\alpha u)\|_\infty \leq (\|x\|_\infty+|\alpha|)\epsilon.^\dagger$$

The bound describes our maximum uncertainty in where the exact shifted vector of abscissae lies, given knowledge only of the computed vector.

It is convenient to think of $\hat{x}$ as a perturbation of the given vector x. The following perturbation bound describes the sensitivity of $\Delta^n\exp_\tau$ to a bounded change in its abscissae.

†Recall that u is a vector of 1's, $u = (1, 1, \ldots, 1)$.

Perturbation bound. Let $x = (\xi_0, \xi_1, \ldots, \xi_n)$ be a vector of abscissae, and $\hat{x} = (\hat{\xi}_0, \hat{\xi}_1, \ldots, \hat{\xi}_n)$ a perturbation of x such that $\max_{0 \le i \le n} |\hat{\xi}_i - \xi_i| \le \gamma\epsilon$ for a constant γ. Then

$$|\Delta''\exp_\tau(\hat{x}) - \Delta''\exp_\tau(x)| \le (e^{\tau\gamma\epsilon} - 1)\cdot\Delta''\exp_\tau(x). \tag{3.6.2}$$

proof: From Theorem 1 in §3.2, $\Delta''\exp_\tau$ is increasing in each of its abscissae, thus

$$\Delta''\exp_\tau(x - \gamma\epsilon u) \le \Delta''\exp_\tau(\hat{x}) \le \Delta''\exp_\tau(x + \gamma\epsilon u).$$

By the translation property (3.1.1),

$$e^{-\tau\gamma\epsilon}\cdot\Delta''\exp_\tau(x) \le \Delta''\exp_\tau(\hat{x}) \le e^{\tau\gamma\epsilon}\cdot\Delta''\exp_\tau(x). \quad \square$$

For small $\tau\gamma\epsilon$ the bound (3.6.2) is equivalent to a relative error of size $\tau\gamma\epsilon$. Hence computational errors may be viewed in the same way as uncertainties in the data. In particular when data uncertainties of size $\gamma\epsilon$ lead to uncertainties of size $\tau\gamma\epsilon$ in the value of $\Delta''\exp_\tau(\hat{x})$ relative to $\Delta''\exp_\tau(x)$, computational errors of comparable, or smaller, size do not greatly increase our uncertainty. Thus, there may be no reason to compute $\Delta''\exp_\tau(x)$ to greater accuracy than about $\tau\gamma\epsilon$. Hence our uncertainty in the data helps answer the question of how much accuracy we are justified in demanding when computing divided differences. We may, then, use the fast standard scheme more in practice, as the data may not warrant using more accurate, but more costly, methods.

Additional modifications of the basic hybrid algorithm may be desireable in practice. For example $\Delta_0''\exp_\tau$ decreases as $\tau''/n!$; so special provisions may be required to represent small numbers during computation. These details, however, must not obscure the important fact about the hybrid algorithm, which is that real exponential divided differences can be computed with high relative accuracy. Such a general statement cannot be made when the abscissae are complex. However, a hybrid type algorithm with error bounds comparable to the above can be developed for some arrangements of complex abscissae. We turn to such a problem in the next three sections.

3.7 Divided differences of the exponential function with complex abscissae.

For applications exponential divided differences with complex abscissae are more important than the real case. In particular a real non-Hermitian matrix A may have complex eigenvalues. These eigenvalues are the abscissae used in forming coefficients of the Newton polynomial form of $\exp(\tau A)$. Therefore it is important to understand which aspects of the real case go over to the complex case, and which do not.

The algorithms presented earlier are applicable to complex abscissae. The theory used to derive the Taylor algorithm, scaling and squaring, and even the hybrid algorithm, depends only upon the exponential function itself. There is no need to distinguish between real and complex data points. Our error bounds and decision criteria, however, do depend explicitly upon the fact that real exponential divided differences are positive. Since complex differences can be zero, we must abandon the idea of strict relative error bounds. Instead, we give error bounds relative to a quantity that bounds or estimates our divided difference.

In this section we examine a few special cases of complex exponential divided differences in order to gain a better understanding of the behavior of such differences. In particular we shall observe how these divided differences are affected by the imaginary parts of the abscissae. Later we indicate how our algorithms may be applied.

We continue studying divided differences of the function $f = \exp_\tau$ with parameter $\tau \geq 0$. Our sequence of abscissae $\mathbf{Z} \equiv \{\zeta_0, \zeta_1, \ldots, \zeta_n, \ldots\}$ may now contain complex elements. We look at three special arrangements of the abscissae. (1) The abscissae lie on a line in the complex plane and are evenly spaced along this line. (2) The sequence of abscissae consists of repetitions of two points, ζ and $-\zeta$; we also look at the case where the two points are conjugates ζ and $\bar{\zeta}$. (3) Finally, we examine the case where the sequence of data points consists exclusively of conjugate pair points. In the first two examples we achieve explicit formulas for the divided differences. In the final case, we characterize the differences by upper bounds on them. This final case is of most interest in matrix function computations because the eigenvalues of real matrices are either real or members of complex conjugate pairs.

Evenly spaced, linear abscissae. On a line the abscissae can be ordered. Let ζ_0 be an extreme data point and let 2δ be the spacing between the abscissae. Then $\mathbf{Z} = \{\zeta_0, \zeta_0+2\delta, \zeta_0+4\delta, \ldots, \zeta_0+2n\delta, \ldots\}$ is the sequence of data points. Exactly as in the real case in §3.2,

$$\Delta_0^n \exp_\tau = \frac{1}{n!} e^{\tau\zeta_0} \left[\frac{e^{2\tau\delta}-1}{2\delta}\right]^n = \frac{1}{n!} e^{\tau(\zeta_0+n\delta)} \left[\frac{\sinh(\tau\delta)}{\delta}\right]^n. \tag{3.7.1}$$

We note that $\Delta_0^n \exp_\tau = 0$ if, and only if, $\tau\delta = k\pi i$ for some non-zero integer k. Thus high order divided differences can be zero. Since two points lie on a line, this also implies that first divided differences are zero if, and only if, their abscissae are separated by $k\pi i$.

Suppose δ is pure imaginary, say $\delta = \nu i$. Let us observe how $|\Delta_0^n \exp_\tau|$ varies with ν. We have

$$|\Delta_0^n \exp_\tau| = \frac{1}{n!} |e^{\tau(\zeta_0+n\nu i)}| \left|\frac{\sinh(\tau\nu i)}{\nu i}\right|^n$$

$$= \frac{1}{n!} e^{\tau\xi_0} \left|\frac{\sin(\tau\nu)}{\nu}\right|^n$$

where $\xi_0 \equiv \mathrm{Re}(\zeta_0)$. $|\Delta_0^n \exp_\tau|$, then, behaves as a damped sine wave, becoming smaller with increasing ν. It has local maxima when $\tan(\tau\nu) = \tau\nu$. For $\xi_0 = 0$ and $\tau = 1$, the table in Fig. 3.7.1 lists some of these maxima for $n = 1, 2, \ldots, 7$.

ν	$\lvert\Delta_0^1\exp\rvert$	$\lvert\Delta_0^2\exp\rvert$	$\lvert\Delta_0^3\exp\rvert$	$\lvert\Delta_0^4\exp\rvert$	$\lvert\Delta_0^5\exp\rvert$	$\lvert\Delta_0^6\exp\rvert$	$\lvert\Delta_0^7\exp\rvert$
0	1.00	5.00E-1	1.67E-1	4.17E-2	8.33E-3	1.39E-3	1.98E-4
4.49	2.17E-1	2.36E-2	1.71E-3	9.28E-5	4.03E-6	1.46E-7	4.53E-9
7.73	1.28E-1	8.24E-3	3.53E-4	1.13E-5	2.91E-7	6.22E-9	1.14E-10
10.90	9.13E-2	4.17E-3	1.27E-4	2.90E-6	5.29E-8	8.06E-10	1.05E-11
14.07	7.09E-2	2.51E-3	5.94E-5	1.05E-6	1.49E-8	1.77E-10	1.79E-12
17.22	5.80E-2	1.68E-3	3.25E-5	4.71E-7	5.46E-9	5.27E-11	4.37E-13
20.37	4.90E-2	1.20E-3	1.96E-5	2.41E-7	2.36E-9	1.93E-11	1.35E-13
23.52	4.25E-2	9.02E-4	1.28E-5	1.36E-7	1.15E-9	8.16E-12	4.95E-14
26.67	3.75E-2	7.02E-4	8.77E-6	8.22E-8	6.16E-10	3.85E-12	2.06E-14
29.81	3.35E-2	5.62E-4	6.28E-6	5.26E-8	3.53E-10	1.97E-12	9.44E-15

Fig. 3.7.1: Maxima of $|\Delta_0^n \exp|$, as a function of ν, for evenly spaced imaginary abscissae.

The magnitude of these divided differences is strongly affected by the difference in the imaginary parts of the abscissae. Our study of complex exponential divided differences must take this into account. The next example even more clearly illustrates this dependence on the imaginary parts.

Two-point exponential divided differences. In an example in §2.8 we saw that divided differences of exponential functions for sequences of data points like $Z = \{\zeta, -\zeta, \zeta, -\zeta, \ldots\}$, where a point and its negative are repeated, have many special properties. In particular, we found that the related functions

$$b_n(\zeta) \equiv \frac{1}{2}\{(\Delta^{2n}\exp_\tau)(\zeta, -\zeta, \ldots, \zeta) + (\Delta^{2n}\exp_\tau)(-\zeta, \zeta, \ldots, -\zeta)\} \tag{3.7.2a}$$

$$a_n(\zeta) \equiv (\Delta^{2n+1}\exp_\tau)(\zeta, -\zeta, \ldots, \zeta, -\zeta) \tag{3.7.2b}$$

satisfy the recurrences

$$b_n(\zeta) = \frac{\tau a_{n-1}(\zeta)}{2n} \tag{3.7.3a}$$

$$a_n(\zeta) = \frac{\tau b_{n-1}(\zeta) - (2n-1)a_{n-1}(\zeta)}{2n\zeta^2} \tag{3.7.3b}$$

for $n = 1, 2, \ldots$, where

$$b_0(\zeta) = \cosh(\tau\zeta) \tag{3.7.4a}$$

$$a_0(\zeta) = \frac{\sinh(\tau\zeta)}{\zeta}. \tag{3.7.4b}$$

From these relations, we show that the functions b_n and a_n are representable in terms of spherical Bessel functions, commonly denoted j_n. In addition, we derive a simple assymptotic expression for the two-point divided difference $(\Delta^{2n}\exp_\tau)(\zeta, -\zeta, \ldots, \zeta)$ as $\tau|\zeta| \to \infty$.

Representation of two-point exponential divided differences. For each $n = 0, 1, 2, \ldots$,

$$b_n(\zeta) = \frac{\tau^{n+1}}{2^n n!(i\zeta)^{n-1}} j_{n-1}(i\tau\zeta) \tag{3.7.5a}$$

$$a_n(\zeta) = \frac{\tau^{n+1}}{2^n n!(i\zeta)^n} j_n(i\tau\zeta), \tag{3.7.5b}$$

where the j_n are spherical Bessel functions. Also as $\tau|\zeta| \to \infty$,

$$(\Delta^{2n}\exp_\tau)(\zeta, -\zeta, \ldots, \zeta) \sim \frac{\tau^n}{2^n n!\zeta^n} e^{\tau\zeta}. \tag{3.7.6}$$

proof: Spherical Bessel functions† are related to the more familiar Bessel functions of the first

†The introduction to the National Bureau of Standards' *Tables of Spherical Bessel Functions* [1947] gives a brief explanation of these functions.

kind, J_m, according to

$$j_n(\omega) = \sqrt{\pi/2\omega}\, J_{n+1/2}(\omega),$$

where the index $m = n+\frac{1}{2}$ indicates a half-order Bessel function. The well-known Bessel recurrence

$$2mJ_m(\omega) = \omega\{J_{m+1}(\omega) + J_{m-1}(\omega)\}$$

becomes

$$(2n+1)j_n(\omega) = \omega\{j_{n+1}(\omega) + j_{n-1}(\omega)\} \tag{3.7.7}$$

for spherical Bessel functions. Initially

$$j_0(\omega) = \frac{\sin(\omega)}{\omega} \quad \text{and} \quad j_{-1}(\omega) = \frac{\cos(\omega)}{\omega}. \tag{3.7.8}$$

When $n=0$ in (3.7.5a-b), a comparison of (3.7.4a-b) with (3.7.8) shows that initially the j_{-1} and j_0 in (3.7.5a-b) are spherical Bessel functions. For general n we derive the Bessel *recurrence* (3.7.7) for j_n from the recurrences (3.7.3a-b) for b_n and a_n. Inserting (3.7.5a-b) into (3.7.3b) yields

$$\frac{\tau^{n+1}}{2^n n!(i\zeta)^n} j_n(i\tau\zeta) = \frac{1}{2n\zeta^2}\left\{\frac{\tau^{n+1}}{2^{n-1}(n-1)!(i\zeta)^{n-2}} j_{n-2}(i\tau\zeta) - \frac{(2n-1)\tau^n}{2^{n-1}(n-1)!(i\zeta)^{n-1}} j_{n-1}(i\tau\zeta)\right\}$$

or

$$j_n(i\tau\zeta) = -j_{n-2}(i\tau\zeta) - \frac{i(2n-1)}{\tau\zeta} j_{n-1}(i\tau\zeta).$$

When this is rearranged and the index n is increased by 1, we obtain (3.7.7) with $\omega = i\tau\zeta$; hence each j_n in (3.7.5a-b) is a spherical Bessel function.

For large $|\omega|$ spherical Bessel functions behave, assymptotically, as

$$j_n(\omega) \sim \frac{1}{\omega}\cos[\omega - (n+1)\pi/2].$$

Thus as $\tau|\zeta| \to \infty$ we have

$$(\Delta^{2n}\exp_\tau)(\zeta, -\zeta, \ldots, \zeta) = b_n(\zeta) + \zeta a_n(\zeta)$$

$$= \frac{\tau^{n+1}}{2^n n!(i\zeta)^{n-1}}\{j_{n-1}(i\tau\zeta) - ij_n(i\tau\zeta)\}$$

$$\sim \frac{\tau^n}{2^n n!(i\zeta)^n}\{\cos(i\tau\zeta - n\pi/2) - i\sin(i\tau\zeta - n\pi/2)\}$$

$$= \frac{\tau^n}{2^n n!\zeta^n} e^{\tau\zeta} \quad \square$$

The translation property (3.1.1) provides an immediate corollary to the above when the sequence of abscissae consists of ζ and $\bar{\zeta}$, repeated.

Corollary: Let the sequence of abscissae $Z = \{\zeta, \bar{\zeta}, \zeta, \bar{\zeta}, \ldots\}$, where $\zeta = \xi + i\eta$ and its conjugate $\bar{\zeta} = \xi - i\eta$ are repeated. Then for each $n = 0, 1, 2, \ldots$,

$$\mathrm{Re}(\Delta_0^{2n}\exp_\tau) = e^{\tau\xi} b_n(i\eta) = \frac{(-1)^{n-1}\tau^{n+1}}{2^n n!\eta^{n-1}} e^{\tau\xi} j_{n-1}(-\tau\eta) \tag{3.7.9a}$$

$$\mathrm{Im}(\Delta_0^{2n}\exp_\tau) = \eta\cdot\Delta_0^{2n+1}\exp_\tau = \eta e^{\tau\xi} a_n(i\eta) = \frac{(-1)^n\tau^{n+1}}{2^n n!\eta^{n-1}} e^{\tau\xi} j_n(-\tau\eta). \tag{3.7.9b}$$

Further as $\eta \to \infty$,

$$\Delta_0^{2n}\exp_\tau \sim \frac{(-i\tau)^n}{2^n n!\eta^n} e^{\tau\zeta}. \tag{3.7.10}$$

proof: By the translation property,

$$\Delta_0^{2n}\exp_\tau \equiv (\Delta^{2n}\exp_\tau)(\zeta, \bar{\zeta}, \ldots, \zeta) = e^{\tau\xi}\cdot(\Delta^{2n}\exp_\tau)(i\eta, -i\eta, \ldots, i\eta).$$

The results follow from (3.7.5a-b) and (3.7.6) by inserting $i\eta$ for ζ, and then multiplying by $e^{\tau\xi}$. $\square$

From (3.7.10),

$$|\Delta_0^{2n}\exp_\tau| \sim \frac{\tau^n}{2^n n!\eta^n} e^{\tau\xi}. \tag{3.7.11}$$

The imaginary part leads to a η^{-n} damping of $\Delta_0^{2n}\exp_\tau$. Also since $j_{n-1}(-\tau\eta)$ and $j_n(-\tau\eta)$ are never simultaneously zero, for all $\tau > 0$ the divided difference $\Delta_0^{2n}\exp_\tau \neq 0$.

Exponential divided differences with conjugate pair abscissae. We now turn to the case of a sequence of abscissae consisting of conjugate pair elements. In particular let $Z = \{\zeta_0, \bar{\zeta}_0, \zeta_1, \bar{\zeta}_1, \ldots, \zeta_n, \bar{\zeta}_n, \ldots\}$ where $\zeta_i = \xi_i + i\eta_i$ and $\bar{\zeta}_i = \xi_i - i\eta_i$ with each $\eta_i > 0$. The following bounds, depending on divided differences of $\exp_\tau$ for the real abscissae ξ_i alone, help to describe the dependence of conjugate pair exponential divided differences on both the real and

imaginary parts of the abscissae.

Bounds for conjugate pair exponential divided differences. Let $Z = \{\zeta_0, \bar{\zeta}_0, \ldots, \zeta_n, \bar{\zeta}_n, \ldots\}$ be a sequence of conjugate pair abscissae. Then for each $n \geq 0$,

$$|\Delta_0^{2n}\exp_\tau| \leq (\prod_{j=0}^{n-1}\eta_j)^{-1}\cdot\Delta_{\xi_0}^{n}\exp_\tau \tag{2.7.12a}$$

and

$$|\Delta_0^{2n+1}\exp_\tau| \leq (\prod_{j=0}^{n}\eta_j)^{-1}\cdot\Delta_{\xi_0}^{n}\exp_\tau . \tag{2.7.12b}$$

proof: The proof is by induction on n. We note first, employing a remark after (2.1.3), that since

$$\Delta_0^{2n+1}\exp_\tau = \frac{(\Delta^{2n}\exp_\tau)(\zeta_0, \bar{\zeta}_0, \ldots, \zeta_{n-1}, \bar{\zeta}_{n-1}, \zeta_n) - (\Delta^{2n}\exp_\tau)(\zeta_0, \bar{\zeta}_0, \ldots, \zeta_{n-1}, \bar{\zeta}_{n-1}, \bar{\zeta}_n)}{\zeta_n - \bar{\zeta}_n}$$

$$= \frac{1}{\eta_n}\mathrm{Im}(\Delta_0^{2n}\exp_\tau) ,$$

(3.7.12b) is an immediate consequence of (3.7.12a). When $n = 0$ and $\tau \geq 0$,

$$|\Delta_0^0\exp_\tau| = |e^{\tau\zeta_0}| = e^{\tau\xi_0} = \Delta_{\xi_0}^0\exp_\tau ,$$

and (3.7.12a) certainly holds. Now let us assume it is true for all orders up to $(2n-2)$. That is, we assume for all $\tau \geq 0$

$$|\Delta_0^{2n-2}\exp_\tau| \leq (\prod_{j=0}^{n-2}\eta_j)^{-1}\cdot\Delta_{\xi_0}^{n-1}\exp_\tau ,$$

and hence

$$|\Delta_0^{2n-1}\exp_\tau| \leq (\prod_{j=0}^{n-1}\eta_j)^{-1}\cdot\Delta_{\xi_0}^{n-1}\exp_\tau .$$

By the recursive integral formula (3.1.5),

$$\Delta_0^{2n}\exp_\tau = e^{\tau\zeta_n}\int_0^\tau e^{-\sigma\zeta_n}\cdot\Delta_0^{2n-1}\exp_\sigma d\sigma .$$

Thus

$$|\Delta_0^{2n}\exp_\tau| \leqslant e^{\tau\xi_n}\int_0^\tau e^{-\sigma\xi_n}|\Delta_0^{2n-1}\exp_\sigma| \, d\sigma$$

$$\leqslant (\prod_{j=0}^{n-1}\eta_j)^{-1}\cdot e^{\tau\xi_n}\int_0^\tau e^{-\sigma\xi_n}\cdot\Delta_{\xi_0}^{n-1}\exp_\sigma \, d\sigma$$

$$= (\prod_{j=0}^{n-1}\eta_j)^{-1}\cdot\Delta_{\xi_0}^{n}\exp_\tau . \quad \square$$

When the real parts ξ_j of the abscissae are equal, that is $\xi_0 = \xi_1 = \cdots = \xi_n$, this corollary follows.

Corollary: Let $Z = \{\xi \pm i\eta_j, j = 0, \ldots, n\}$. Then for each $n \geqslant 0$,

$$|\Delta_0^{2n}\exp_\tau| \leqslant (\prod_{j=0}^{n-1}\eta_j)^{-1}\frac{\tau^n e^{\tau\xi}}{n!} \tag{3.7.13a}$$

and

$$|\Delta_0^{2n+1}\exp_\tau| \leqslant (\prod_{j=0}^{n}\eta_j)^{-1}\frac{\tau^n e^{\tau\xi}}{n!} . \tag{3.7.13b}$$

With the exception of the factor 2^n, the bounds in our corollary resemble our assymptotic results for two-point conjugate pair divided differences. This leads us to suspect that

$$2^{-n}(\prod_{j=0}^{n-1}\eta_j)^{-1}\cdot\Delta_{\xi_0}^{n}\exp_\tau \text{ and } 2^{-n}(\prod_{j=0}^{n}\eta_j)^{-1}\cdot\Delta_{\xi_0}^{n}\exp_\tau \tag{3.7.14}$$

are reasonable estimates of $|\Delta_0^{2n}\exp_\tau|$ and $|\Delta_0^{2n+1}\exp_\tau|$, respectively, when the η_j are large. The values in Fig. 3.7.2 illustrate this. Note that not every estimated value by (3.7.14) is large enough to be a bound.

General complex exponential divided differences. When we are unable to make assumptions about the abscissae, we can say little about the behavior of the divided differences. Just as the simple bound derived from (2.1.12),

$$|\Delta_0^n\exp_\tau| \leqslant \frac{1}{n!}\max_{0\leqslant j\leqslant n}|\tau^n e^{\tau\zeta_j}| = \frac{\tau^n}{n!}e^{\tau\xi_n},$$

poorly describes the behavior of $\Delta_0^n\exp_\tau$ when $\xi_n - \xi_0$ is large, even when all the abscissae are real, a bound depending only on the real parts of complex abscissae poorly describes $|\Delta_0^n\exp_\tau|$ when some data points have large imaginary parts. All complex exponential divided differences

n	ζ_n	$\lvert\Delta_0^n \exp\rvert$	bounds from (3.7.12a-b)	estimates from (3.7.14)
0	(0,+10*i*)	1.00	1.00	1.00
1	(0,-10*i*)	5.44E-2	1.00E-1	1.00E-1
2	(1,+9*i*)	8.39E-2	1.72E-1	8.59E-2
3	(1,-9*i*)	9.31E-3	1.91E-2	9.55E-3
4	(2,+11*i*)	3.64E-3	1.64E-2	4.10E-3
5	(2,-11*i*)	2.71E-4	1.49E-3	3.73E-4
6	(3,+10*i*)	9.18E-5	8.54E-4	1.07E-4
7	(3,-10*i*)	3.16E-6	8.54E-5	1.07E-5
8	(4,+9*i*)	2.41E-6	3.67E-5	2.29E-6
9	(4,-9*i*)	2.50E-7	4.08E-6	2.55E-7
10	(5,+10*i*)	4.20E-8	1.40E-6	4.38E-8
11	(5,-10*i*)	1.96E-9	1.40E-7	4.38E-9

Fig. 3.7.2: Bounds and estimates for $\lvert\Delta_0^n \exp\rvert$ with conjugate pair abscissae.

do satisfy the following bound regardless of the imaginary parts of the data points.

Upper bound on $\lvert\Delta_0^n \exp_\tau\rvert$ for complex abscissae. Let $\mathbf{Z} = \{\zeta_0, \zeta_1, \ldots, \zeta_n, \ldots\}$ be a sequence of complex valued abscissae, and let $\xi_j = \mathrm{Re}(\zeta_j)$ for each $j = 0, 1, \ldots, n, \ldots$. Then

$$\lvert\Delta_0^n \exp_\tau\rvert \leqslant \Delta_{\xi_0}^n \exp_\tau \tag{3.7.15}$$

for all $n \geqslant 0$.

proof: Directly from (3.1.3), namely

$$\Delta_0^n \exp_\tau = \int_0^\tau \int_0^{\sigma_1} \cdots \int_0^{\sigma_{n-1}} \exp[\tau\zeta_0 + (\zeta_1 - \zeta_0)\sigma_1 + \cdots + (\zeta_n - \zeta_{n-1})\sigma_n]\, d\sigma_n \cdots d\sigma_2 d\sigma_1 ,$$

we have

$$\lvert\Delta_0^n \exp_\tau\rvert \leqslant \int_0^\tau \int_0^{\sigma_1} \cdots \int_0^{\sigma_{n-1}} \exp[\tau\xi_0 + (\xi_1 - \xi_0)\sigma_1 + \cdots + (\xi_n - \xi_{n-1})\sigma_n]\, d\sigma_n \cdots d\sigma_2 d\sigma_1$$

$$= \Delta_{\xi_0}^n \exp_\tau . \quad \square$$

Comparing (3.7.15) with our conjugate pair bounds shows that the imaginary parts of the abscissae may be very important and should be reflected in any bounds we use. In the next

sections we apply our divided difference algorithms to conjugate pair abscissae and use the upper bounds and estimates we have presented here to derive error bounds for the computations.

3.8 Divided difference tables with conjugate pair abscissae.

The elements of divided difference tables whose abscissae are conjugate pairs are quite special in that some are real and some are conjugates of others. These properties do not depend upon the exponential function, but apply to any function f symmetric about the real axis. For this reason we digress in this section from our study of $\exp_r$ and revert to the function f, where $f(\bar{\zeta}) = \overline{f(\zeta)}$. The results are more general and the notation is more compact. Applications to $\exp_r$ follow in the next section.

$$\begin{vmatrix}
f(\zeta_{-3}) & \Delta^{1}_{-3}f & \Delta^{2}_{-3}f & \Delta^{3}_{-3}f & \Delta^{4}_{-3}f & \Delta^{5}_{-3}f & \Delta^{6}_{-3}f & \underline{\Delta^{7}_{-3}f} \\
 & f(\zeta_{-2}) & \Delta^{1}_{-2}f & \Delta^{2}_{-2}f & \Delta^{3}_{-2}f & \Delta^{4}_{-2}f & \underline{\Delta^{5}_{-2}f} & \underline{\Delta^{6}_{-2}f} \\
 & & f(\zeta_{-1}) & \Delta^{1}_{-1}f & \Delta^{2}_{-1}f & \underline{\Delta^{3}_{-1}f} & \underline{\Delta^{4}_{-1}f} & \Delta^{5}_{-1}f \\
 & & & f(\zeta_{-0}) & \underline{\Delta^{1}_{-0}f} & \underline{\Delta^{2}_{-0}f} & \Delta^{3}_{-0}f & \Delta^{4}_{-0}f \\
 & & & & \underline{f(\zeta_0)} & \Delta^{1}_{0}f & \Delta^{2}_{0}f & \Delta^{3}_{0}f \\
 & & & & & f(\zeta_1) & \Delta^{1}_{1}f & \Delta^{2}_{1}f \\
 & & & & & & f(\zeta_2) & \Delta^{1}_{2}f \\
 & & & & & & & f(\zeta_3)
\end{vmatrix}$$

Fig. 3.8.1: Rearranged divided difference table Δf for $\{\zeta_{-3}, \zeta_{-2}, \zeta_{-1}, \zeta_{-0}, \zeta_0, \zeta_1, \zeta_2, \zeta_3\}$.

We have seen that abscissae should be ordered so that close values are adjacent to each other. It follows that $\bar{\zeta}$ should not be adjacent to ζ when $\mathrm{Im}(\zeta)$ is large, as would be natural. A good, but unorthodox, ordering for complex conjugate pairs of abscissae is $\mathbf{Z} = \{\bar{\zeta}_n, \bar{\zeta}_{n-1}, \ldots, \bar{\zeta}_1, \bar{\zeta}_0, \zeta_0, \zeta_1, \ldots, \zeta_n\}$. Some extra dividends follow from this choice as we show below. In order to maintain (as closely as possible) our notation $\Delta^{k}_{j}f$ to indicate the use of $\zeta_j, \zeta_{j+1}, \ldots, \zeta_{j+k}$, we write ζ_{-j} for $\bar{\zeta}_j$. The table in Fig. 3.8.1 shows a typical Δf where $n = 3$. The entries corresponding to the top row in a naturally ordered table are underlined. These are the entries that are used, for example, as coefficients in a Newton polynomial.

Let Z be the "step matrix" associated with the sequence $\mathbf{Z}$, that is

$$Z = \begin{vmatrix} \zeta_{-n} & 1 & & & & & & & & & \\ & \zeta_{-n+1} & 1 & & & & & & & & \\ & & \ddots & \ddots & & & & & & & \\ & & & \ddots & \ddots & & & & & & \\ & & & & \zeta_{-1} & 1 & & & & & \\ & & & & & \zeta_{-0} & 1 & & & & \\ & & & & & & \zeta_0 & 1 & & & \\ & & & & & & & \zeta_1 & \ddots & & \\ & & & & & & & & \ddots & \ddots & \\ & & & & & & & & & \ddots & 1 \\ & & & & & & & & & & \zeta_n \end{vmatrix}$$

This extra property referred to above is that both Z and $\Delta f = f(Z)$ are Hermitian about the secondary diagonal (bottom left to top right). This property is most conveniently expressed in terms of the permutation matrix

$$\hat{I} = \begin{vmatrix} & & & & & & & 1 \\ & & & & & & 1 & \\ & & & & & \iddots & & \\ & & & & \iddots & & & \\ & & & 1 & & & & \\ & & \iddots & & & & & \\ & 1 & & & & & & \end{vmatrix}$$

Secondary symmetry of Δf. If $Z = \{\bar{\zeta}_n, \ldots, \bar{\zeta}_0, \zeta_0, \ldots, \zeta_n\}$ and $f(\bar{\zeta}) = \overline{f(\zeta)}$, then both $\hat{I} \cdot Z$ and $\hat{I} \cdot \Delta f$ are Hermitian.

proof: Any complex matrix B is Hermitian if $\overline{B^T} = B$. We denote the conjugate transpose of B by B^*. By inspection, $\hat{I} \cdot Z$ is Hermitian,

$$\hat{I} \cdot Z = Z^* \cdot \hat{I}.$$

From the conjugate transpose property (1.1.6),

$$f(Z^*) = f(Z)^*;$$

so employing (1.1.3) for similarity transformations,

$$f(Z)^*\cdot\hat{I} = f(Z^*)\cdot\hat{I} = \hat{I}\cdot f(\hat{I}Z^*\hat{I})$$

$$= \hat{I}\cdot f(\hat{I}\cdot\hat{I}Z) = \hat{I}\cdot f(Z),$$

and $\hat{I}\cdot f(Z)$ is Hermitian. □

This result means every divided difference lying on the secondary diagonal of $\mathbf{\Delta} f$ is real, and every divided difference below this diagonal has a conjugate above the diagonal. For example in Fig. 3.8.1, $\mathbf{\Delta}^1_{-0}f$, $\mathbf{\Delta}^3_{-1}f$, $\mathbf{\Delta}^5_{-2}f$ and $\mathbf{\Delta}^7_{-3}f$ are all real, and $\mathbf{\Delta}^1_{-3}f = \overline{\mathbf{\Delta}^1_{-2}f}$ while $\mathbf{\Delta}^3_{-2}f = \overline{\mathbf{\Delta}^3_{-0}f}$. Only the portion of the table on and below the secondary diagonal ever need be computed. For example, $\mathbf{\Delta}^3_{0}f$ and all differences upon which it depends might be computed by a series method because the abscissae may be close. The standard formula and taking conjugates will fill out the rest of the table. The idea is illustrated in Fig. 3.9.1.

From our discussion here, the reordered table is clearly ideal for computation by a hybrid algorithm. We consider this for $f = \exp$, in the next section.

example: Fig. 3.8.2 shows that reordering abscissae and computing first order differences by a special formula may have a dramatic effect on error propagation when the standard scheme is used. The abscissae here are

$$\zeta_{\pm 0} = -1.414214 \pm i8.585786$$
$$\zeta_{\pm 1} = 1.412799 \pm i11.41563$$
$$\zeta_{\pm 2} = 1.414214 \pm i11.41421$$
$$\zeta_{\pm 3} = 1.417039 \pm i11.41138$$

First order (initial) differences were computed correct to seven decimal digits. The standard scheme was employed, thereafter, in greater precision to isolate propagated errors. The figure compares divided differences from the top row of the table for the natural ordering of the data points with the identical differences when the data points are reordered as suggested in this section.

Reordering permits many differences, for which error growth would be large by the standard scheme, to be computed by a special method. We see here with reordering that close abscissae contribute only to first and second order differences. These first order differences do not contribute to error growth when computed specially. However, failure to compute first

Order n	Correct values to 7 digits	Natural ordering with special computation of first differences
0	(-1.624537E-1, 1.808715E-1)	(-1.624537E-1, 1.808715E-1)
1	(2.106638E-2, 0.0)	(2.106638E-2, 0.0)
2	(-5.269139E-2, 1.213605E-2)	(-5.269140E-2, 1.213604E-2)
3	(1.063108E-3, 0.0)	(1.063107E-3, 3.965390E-10)
4	(-1.114105E-4, 1.907577E-3)	(-1.113896E-4, 1.907435E-3)
5	(1.671230E-4, 0.0)	(1.671291E-4,-2.643648E-9)
6	(3.230809E-5, 1.838758E-5)	(3.215634E-5, 1.820492E-5)
7	(1.611337E-6, 0.0)	(1.547398E-6,-3.597914E-8)
Order n	**Reordering with special computation of first differences**	**Reordering without special computation of first differences**
0	(-1.624537E-1, 1.808715E-1)	(-1.624537E-1, 1.808715E-1)
1	(2.106638E-2, 0.0)	(2.106639E-2, 0.0)
2	(-5.269141E-2, 1.213605E-2)	(-5.269139E-2, 1.213604E-2)
3	(1.063108E-3, 0.0)	(1.063108E-3, 0.0)
4	(-1.114107E-4, 1.907577E-3)	(-1.113844E-4, 1.907667E-3)
5	(1.671230E-4, 0.0)	(1.671309E-4, 0.0)
6	(3.230604E-5, 1.838538E-5)	(3.203465E-5, 1.802915E-5)
7	(1.611144E-6, 0.0)	(1.579928E-6, 0.0)

Fig. 3.8.2: Effects of reordering data points and special computation of first divided differences on $\Delta''\exp$.

order differences accurately destroys any benefit from reordering, as the numbers show.

When clusters of close abscissae are small, as here, reordering the abscissae makes special computation of low order differences very effective in controlling error growth. In the next section we shall see that a hybrid algorithm effects even more dramatic improvements in accuracy.

3.9 A hybrid algorithm for $\Delta\exp_r$ with conjugate pair abscissae.

The table rearrangement presented in the last section strongly suggests implementing a hybrid algorithm for computing $\Delta\exp_r$ when the abscissae are conjugate pairs. A hybrid algorithm using scaling and squaring, as well as the standard scheme, is most accurate for abscissae having imaginary parts nearly equal in absolute value, but large. Fig. 3.9.1 illustrates the conjugate symmetry relationships in a reordered conjugate pair divided difference table. The "s" indicates an element which may be computed by scaling and squaring, "x" by the standard scheme, while "r" means the element is real.

$$\begin{vmatrix} \bar{s} & \bar{s} & \bar{s} & \bar{s} & \bar{x} & \bar{x} & \bar{x} & r \\ & \bar{s} & \bar{s} & \bar{s} & \bar{x} & \bar{x} & r & x \\ & & \bar{s} & \bar{s} & \bar{x} & r & x & x \\ & & & \bar{s} & r & x & x & x \\ & & & & s & s & s & s \\ & & & & & s & s & s \\ & & & & & & s & s \\ & & & & & & & s \end{vmatrix}$$

Fig. 3.9.1: Relation of entries in conjugate pair table.

$$\Delta\exp_r = \begin{vmatrix} \overline{e^{r\zeta_3}} & \overline{\Delta_2^1} & \overline{\Delta_1^2} & \overline{\Delta_0^3} & \overline{\Delta_{-0}^4} & \overline{\Delta_{-1}^5} & \overline{\Delta_{-2}^6} & \underline{\Delta_{-3}^7} \\ & \overline{e^{r\zeta_2}} & \overline{\Delta_1^1} & \overline{\Delta_0^2} & \overline{\Delta_{-0}^3} & \overline{\Delta_{-1}^4} & \underline{\Delta_{-2}^5} & \Delta_{-2}^6 \\ & & \overline{e^{r\zeta_1}} & \overline{\Delta_0^1} & \overline{\Delta_{-0}^2} & \underline{\Delta_{-1}^3} & \Delta_{-1}^4 & \Delta_{-1}^5 \\ & & & \overline{e^{r\zeta_0}} & \Delta_{-0}^1 & \Delta_{-0}^2 & \Delta_{-0}^3 & \Delta_{-0}^4 \\ & & & & e^{r\zeta_0} & \Delta_0^1 & \Delta_0^2 & \Delta_0^3 \\ & & & & & e^{r\zeta_1} & \Delta_1^1 & \Delta_1^2 \\ & & & & & & e^{r\zeta_2} & \Delta_2^1 \\ & & & & & & & e^{r\zeta_3} \end{vmatrix}$$

Fig. 3.9.2: Conjugate pair divided difference table showing symmetries.

Algorithm: Hybrid algorithm for conjugate pair abscissae. For conjugate pair abscissae $\zeta_j = \xi_j + i\eta_j$ and $\zeta_{-j} \equiv \bar{\zeta}_j = \xi_j - i\eta_j$ with $\eta_j >> 0$, $j = 0, 1, \ldots, n$, form the divided difference table matrix as follows:

1. Reorder the sequence of data points as $\{\zeta_{-n}, \ldots, \zeta_{-1}, \zeta_{-0}, \zeta_0, \zeta_1, \ldots, \zeta_n\}$ and, if possible (by reindexing if necessary), so that $\xi_0 \leqslant \xi_1 \leqslant \cdots \leqslant \xi_n$.
2. Compute $\Delta_0^{\prime\prime}\exp_\tau$, and hence each $\Delta_j^k\exp_\tau$ for $j = 0, 1, \ldots, n$ and $k = 0, 1, \ldots, n-j$, by scaling and squaring (§3.4).
3. For each $j = 0, 1, \ldots, n$ compute ($-j+1 \equiv 0$ when $j = 0$)
   a. on the secondary diagonal of the table
   $$\Delta_{-j}^{2j+1}\exp_\tau = \frac{1}{\eta_j}\mathrm{Im}(\Delta_{-j+1}^{2j}\exp_\tau);$$
   b. for $k = 2j+2, \ldots, j+n+1$ each $\Delta_{-j}^{k}\exp_\tau$ by the standard scheme, e.g.
   $$\Delta_{-j}^{k}\exp_\tau = \frac{\Delta_{-j+1}^{k-1}\exp_\tau - \Delta_{-j}^{k-1}\exp_\tau}{\zeta_{k-j-1} - \zeta_{-j}}.$$
4. Fill the remainder of the table using conjugate symmetry about the secondary diagonal.

When $n = 3$ the matrix in Fig. 3.9.2 illustrates the relation between various elements in a table. (Some references to the function $\exp_\tau$ are suppressed.) In the hybrid algorithm entries below the horizontal line in Fig. 3.9.2 are computed by scaling and squaring (step 2), while entries to the left of the vertical line are just conjugates of these, as indicated.

Next in step 3a of the algorithm,

$$\Delta_{-0}^{1}\exp_\tau = \frac{\exp_\tau(\zeta_0) - \exp_\tau(\bar{\zeta}_0)}{\zeta_0 - \bar{\zeta}_0} = \frac{1}{\eta_0}\mathrm{Im}[\exp_\tau(\zeta_0)] = e^{\tau\xi_0}\frac{\sin(\tau\eta_0)}{\eta_0}.$$

This and the row immediately below, already known from the scaling and squaring step, permit completion of the -0 table row by the standard scheme (step 3b). Then again in step 3a,

$$\Delta_{-1}^{3}\exp_\tau = \frac{1}{\eta_1}\mathrm{Im}(\Delta_{-0}^{2}\exp_\tau),$$

and the elements to the right of $\Delta_{-1}^{3}\exp_\tau$ are computed by the standard formula (step 3b). For example

$$\blacktriangle_{-1}^{5}\exp_r = \frac{\blacktriangle_{-0}^{4}\exp_r - \blacktriangle_{-1}^{4}\exp_r}{\zeta_2 - \zeta_{-1}}.$$

Step 3 is, then, repeated until $\blacktriangle_{-3}^{7}\exp_r$ is computed. The remaining elements to the right of the vertical line are conjugates of elements above the horizontal line, as indicated. The elements on the secondary diagonal (indicated by underlining in Fig. 3.9.2), computed in step 3a, are all real.

$$\left|\begin{array}{ccccccc}
e^{\tau\xi_3} & \blacktriangle_{\xi_2}^{1} & \blacktriangle_{\xi_1}^{2} & \blacktriangle_{\xi_0}^{3} & \frac{1}{\eta_0}\blacktriangle_{\xi_0}^{3} & \frac{1}{\eta_0\eta_1}\blacktriangle_{\xi_0}^{3} & \frac{1}{\eta_0\eta_1\eta_2}\blacktriangle_{\xi_0}^{3} & \frac{1}{\eta_0\eta_1\eta_2\eta_3}\blacktriangle_{\xi_0}^{3} \\
 & e^{\tau\xi_2} & \blacktriangle_{\xi_1}^{1} & \blacktriangle_{\xi_0}^{2} & \frac{1}{\eta_0}\blacktriangle_{\xi_0}^{2} & \frac{1}{\eta_0\eta_1}\blacktriangle_{\xi_0}^{2} & \frac{1}{\eta_0\eta_1\eta_2}\blacktriangle_{\xi_0}^{2} & \frac{1}{\eta_0\eta_1\eta_2}\blacktriangle_{\xi_0}^{3} \\
 & & e^{\tau\xi_1} & \blacktriangle_{\xi_0}^{1} & \frac{1}{\eta_0}\blacktriangle_{\xi_0}^{1} & \frac{1}{\eta_0\eta_1}\blacktriangle_{\xi_0}^{1} & \frac{1}{\eta_0\eta_1}\blacktriangle_{\xi_0}^{2} & \frac{1}{\eta_0\eta_1}\blacktriangle_{\xi_0}^{3} \\
 & & & e^{\tau\xi_0} & \frac{1}{\eta_0}e^{\tau\xi_0} & \frac{1}{\eta_0}\blacktriangle_{\xi_0}^{1} & \frac{1}{\eta_0}\blacktriangle_{\xi_0}^{2} & \frac{1}{\eta_0}\blacktriangle_{\xi_0}^{3} \\
 & & & & e^{\tau\xi_0} & \blacktriangle_{\xi_0}^{1} & \blacktriangle_{\xi_0}^{2} & \blacktriangle_{\xi_0}^{3} \\
 & & & & & e^{\tau\xi_1} & \blacktriangle_{\xi_1}^{1} & \blacktriangle_{\xi_1}^{2} \\
 & & & & & & e^{\tau\xi_2} & \blacktriangle_{\xi_2}^{1} \\
 & & & & & & & e^{\tau\xi_3}
\end{array}\right|$$

Fig. 3.9.3: Table of upper bounds based on real divided differences.

Upper bounds. For an error analysis of the hybrid algorithm we must first develop error bounds on the scaling and squaring portion of the computation. Then we can see how these errors are propagated during the remainder of the computation by the standard formula. An examination of the upper bounds (3.7.12a-b) and (3.7.15) yields quantities relative to which we may construct error bounds. For example, the table in Fig. 3.9.2 is bounded, element by element, by the table in Fig 3.9.3. Here we omit the function reference $\exp_r$, for clarity, and point out that the divided differences in Fig. 3.9.3 are for the real abscissae $\{\xi_0, \xi_1, \ldots, \xi_n\}$. Our error bounds will be relative to an upper bound matrix such as the one in Fig. 3.9.3.

$$
\begin{vmatrix}
e^{\tau\epsilon_1} & \Delta^1_{\epsilon_2} & \Delta^2_{\epsilon_1} & \Delta^3_{\epsilon_0} & \frac{1}{2\eta_0}\Delta^3_{\epsilon_0} & \frac{1}{4\eta_0\eta_1}\Delta^3_{\epsilon_0} & \frac{1}{8\eta_0\eta_1\eta_2}\Delta^3_{\epsilon_0} & \frac{1}{8\eta_0\eta_1\eta_2\eta_3}\Delta^3_{\epsilon_0} \\
 & e^{\tau\epsilon_2} & \Delta^1_{\epsilon_1} & \Delta^2_{\epsilon_0} & \frac{1}{2\eta_0}\Delta^2_{\epsilon_0} & \frac{1}{4\eta_0\eta_1}\Delta^2_{\epsilon_0} & \frac{1}{4\eta_0\eta_1\eta_2}\Delta^2_{\epsilon_0} & \frac{1}{8\eta_0\eta_1\eta_2}\Delta_{\epsilon_0} \\
 & & e^{\tau\epsilon_1} & \Delta^1_{\epsilon_0} & \frac{1}{2\eta_0}\Delta^1_{\epsilon_0} & \frac{1}{2\eta_0\eta_1}\Delta^1_{\epsilon_0} & \frac{1}{4\eta_0\eta_1}\Delta^2_{\epsilon_0} & \frac{1}{4\eta_0\eta_1}\Delta^3_{\epsilon_0} \\
 & & & e^{\tau\epsilon_0} & \frac{1}{\eta_0}e^{\tau\epsilon_0} & \frac{1}{2\eta_0}\Delta^1_{\epsilon_0} & \frac{1}{2\eta_0}\Delta^2_{\epsilon_0} & \frac{1}{2\eta_0}\Delta^3_{\epsilon_0} \\
 & & & & e^{\tau\epsilon_0} & \Delta^1_{\epsilon_0} & \Delta^2_{\epsilon_0} & \Delta^3_{\epsilon_0} \\
 & & & & & e^{\tau\epsilon_1} & \Delta^1_{\epsilon_1} & \Delta^2_{\epsilon_1} \\
 & & & & & & e^{\tau\epsilon_2} & \Delta^1_{\epsilon_2} \\
 & & & & & & & e^{\tau\epsilon_3}
\end{vmatrix}
$$

Fig. 3.9.4: Table of estimated absolute values.

When the η_j are nearly equal, but large, the error may be estimated, element by element, by $\epsilon\kappa$ times the matrix in Fig. 3.9.4. The constant κ depends only on errors introduced in the scaling and squaring part of the algorithm. This result is not surprising. When the η_j are large, the standard scheme is employed only for well-separated abscissae. From our earlier studies there is little error growth in this case. Indeed, such separation of the data points is the reason for reordering them, in the first place.

example: With the data from the example at the end of §3.8, namely

$$\zeta_{\pm 0} = -1.414214 \pm i8.585786$$
$$\zeta_{\pm 1} = 1.412799 \pm i11.41563$$
$$\zeta_{\pm 2} = 1.414214 \pm i11.41421$$
$$\zeta_{\pm 3} = 1.417039 \pm i11.41138,$$

the tables in Fig. 3.9.5 show that upper bounds, as in Fig. 3.9.3, and estimated absolute values, as in Fig. 3.9.4, describe the size of the divided differences. From symmetry, only the portion of each table on and below the secondary diagonal is shown. The divided differences themselves are listed in Fig. 3.9.6.

Scaling and squaring error bounds. Error bounds from our earlier analysis of scaling and squaring in §3.4 carry over immediately to that portion of the conjugate pair table computed by this method. The bounds are no longer valid relative to the computed difference itself, but rather to an appropriate upper bound on this difference. A quick reexamination of the

Layout of tables				Correct absolute values			
·	·	·	$\Delta^{7}_{-3}\exp$	·	·	·	1.61E-6
·	·	$\Delta^{5}_{-2}\exp$	$\Delta^{6}_{-2}\exp$	·	·	1.67E-4	3.72E-5
·	$\Delta^{3}_{-1}\exp$	$\Delta^{4}_{-1}\exp$	$\Delta^{5}_{-1}\exp$	·	1.06E-3	1.91E-3	7.80E-4
$\Delta^{1}_{-0}\exp$	$\Delta^{2}_{-0}\exp$	$\Delta^{3}_{-0}\exp$	$\Delta^{4}_{-0}\exp$	2.11E-2	5.41E-2	4.26E-2	1.74E-2
e^{ζ_0}	$\Delta^{1}_{0}\exp$	$\Delta^{2}_{0}\exp$	$\Delta^{4}_{0}\exp$	2.43E-1	1.08	8.54E-1	3.53E-1
	e^{ζ_1}	$\Delta^{1}_{1}\exp$	$\Delta^{2}_{1}\exp$		4.11	4.11	2.06
		e^{ζ_2}	$\Delta^{1}_{2}\exp$			4.11	4.12
			e^{ζ_3}				4.12
Upper bounds (Fig. 3.9.3)				Estimated values (Fig. 3.9.4)			
·	·	·	3.01E-5	·	·	·	3.76E-6
·	·	8.67E-4	3.43E-4	·	·	2.17E-4	4.29E-5
·	1.39E-2	9.90E-3	3.92E-3	·	6.97E-3	2.47E-3	9.80E-4
2.83E-2	1.59E-1	1.13E-1	4.47E-2	2.83E-2	7.96E-2	5.65E-2	2.24E-2
2.43E-1	1.37	9.70E-1	3.84E-1	2.43E-1	1.37	9.70E-1	3.84E-1
	4.11	4.11	2.06		4.11	4.11	2.06
		4.11	4.12			4.11	4.12
			4.12				4.12

Fig. 3.9.5: Example of bounds and estimates for conjugate pair divided differences in $\Delta\exp$.

derivation of scaling and squaring error bounds will show this. We study only the double precision accumulation case, as the argument is exactly the same in the single precision case.

Scaling and squaring bound. Consider only the non-conjugate portion of the sequence of abscissae, namely $\{\zeta_0, \zeta_1, \ldots, \zeta_n\}$. For double precision accumulation of inner products,

$$|fl_2(\Delta\exp_\tau) - \Delta\exp_\tau| \leqslant \epsilon[\kappa_2\tau\theta - 1]\cdot\Delta_X\exp_\tau, \tag{3.9.1}$$

where $\kappa_2 = 8.3259$, θ is the maximum spread in the abscissae, and $\Delta_X\exp_\tau$ is the related divided difference table whose abscissae are $X = \{\mathrm{Re}(\zeta_j),\ j = 0, \ldots, n\}$. For single precision accumulation,

$$|fl(\Delta\exp_\tau) - \Delta\exp_\tau| \leqslant \epsilon(n+1)[\kappa_1\tau\theta - 1]\cdot\Delta_X\exp_\tau, \tag{3.9.2}$$

where $\kappa_1 = 21.2950$.

proof: We first compute a scaled divided difference table by the Taylor algorithm. The expansion point α may be the center of the smallest circle enclosing the data points, and the spread θ is the diameter of that circle. Let the data points be ordered so that the real parts satisfy $\xi_0 \leqslant \xi_1 \leqslant \cdots \leqslant \xi_n$. Because exponential divided differences with real abscissae are increasing in each abscissa, we have

$$\frac{\tau^n e^{\tau\xi_0}}{n!} \leqslant \Delta^n_{\xi_0}\exp_\tau$$

and

$$\frac{\tau^n|e^{\tau\alpha}|}{n!} \leqslant \frac{\tau^n}{n!}e^{\tau(\xi_0+\theta/2)} = e^{\tau\theta/2}\cdot\frac{\tau^n e^{\tau\xi_0}}{n!} \leqslant e^{\tau\theta/2}\cdot\Delta^n_{\xi_0}\exp_\tau.$$

In Appendix B we derive the error bound

$$|fl_2(\Delta_0^n\exp_\tau) - \Delta_0^n\exp_\tau| \leqslant \epsilon(2+\tau\theta/2)e^{\tau\theta/2}\frac{\tau^n|e^{\tau\alpha}|}{n!}.$$

Therefore,

$$|fl_2(\Delta_0^n\exp_\tau) - \Delta_0^n\exp_\tau| \leqslant \epsilon(2+\tau\theta/2)e^{\tau\theta}\cdot\Delta^n_{\xi_0}\exp_\tau. \tag{3.9.3}$$

The Taylor series error bound (3.9.3) applies to every element of the divided difference table. The error in the original scaled matrix in scaling and squaring, then, must satisfy the matrix inequality

$$|fl_2(\Delta\exp_{2^{-j}\tau}) - \Delta\exp_{2^{-j}\tau}| \leqslant \epsilon(2+2^{-(j+1)}\tau\theta)e^{2^{-j}\tau\theta}\cdot\Delta_X\exp_{2^{-j}\tau}. \tag{3.9.4}$$

where 2^{-j} is the scaling factor.† The subscript X indicates the divided difference table matrix $\Delta_X \exp_{2^{-j}\tau}$ has real abscissae $X = \{\xi_0, \xi_1, \ldots, \xi_n\}$.

For

$$\beta_j \equiv (2 + 2^{-(j+1)}\tau\theta)e^{2^{-j}\tau\theta},$$

the error in each element of $fl_2(\Delta \exp_{2^{-j}\tau})$ is bounded by $\epsilon\beta_j$ times the corresponding element of $\Delta_X \exp_{2^{-j}\tau}$. This β_j is exactly that used earlier in §3.4. For any complex matrix B

$$|fl_2(B^2) - B^2| \leqslant \epsilon |B|^2.$$

Also bound (3.7.15) yields

$$|\Delta \exp_{2^{-j}\tau}| \leqslant \Delta_X \exp_{2^{-j}\tau};$$

hence,

$$|\Delta \exp_{2^{-j}\tau}|^2 \leqslant \Delta_X \exp_{2^{-(j-1)}\tau}.$$

The same argument that led up to (3.4.9) gives

$$|fl_2(\Delta \exp_\tau) - \Delta \exp_\tau| \leqslant \epsilon[2^j(\beta_j + 1) - 1] \cdot \Delta_X \exp_\tau,$$

where $2^j(\beta_j + 1) - 1$ is the same as in (3.4.9). It is minimized in the same way. For j the smallest non-negative integer satisfying (3.4.10), namely

$$2^{-j}\tau\theta \leqslant 1.3292,$$

we obtain

$$|fl_2(\Delta \exp_\tau) - \Delta \exp_\tau| \leqslant \epsilon[\kappa_2 \tau\theta - 1] \cdot \Delta_X \exp_\tau$$

where $\kappa_2 = 8.3259$. The same argument shows the single precision bound is (3.4.15). □

---

†The matrix bound (3.9.4) does not hold, rigorously, when $\Delta \exp_{2^{-j}\tau}$ is backfilled from its top row. This is because

$$\Delta^{k-1}_{\xi_{i+1}} \exp_\tau < \Delta^{k-1}_{\xi_i} \exp_\tau + |\zeta_{i+k} - \zeta_i| \cdot \Delta^k_{\xi_i} \exp_\tau$$

when $\eta_{i+k} \neq \eta_i$. If the Taylor formula is used on the entire table, (3.9.4) does hold.

Relations (3.9.1) and (3.9.2) mean that the error in complex exponential divided differences, relative to corresponding real divided differences, has the same bound for scaling and squaring as in the exclusively real case.

Standard scheme error bounds. When all the imaginary parts η_j, $j=0,1,\ldots,n$, are large and nearly equal, the portion of the table computed according to the standard scheme satisfies the following error bounds.

Standard scheme error bound. When each η_j, $j=0,1,\ldots,n$, is large compared with n/τ,

$$|fl(\Delta_{-k}^{2k+l}\exp_\tau) - \Delta_{-k}^{2k+l}\exp_\tau| \leqslant \epsilon\kappa\Big(\prod_{j=0}^{k}\eta_j\Big)^{-1}\cdot\Delta_{\xi_0}^{k+l-1}\exp_\tau \tag{3.9.5}$$

for each $k=0,1,\ldots,n$ and $l=1,2,\ldots,n-k+1$. κ is one of the scaling and squaring error coefficients $\kappa_2\tau\theta-1$ or $(n+1)[\kappa_1\tau\theta-1]$, depending on the arithmetic used.

The recursive nature of the standard scheme makes it easiest to describe bounds on the propagation and growth of errors in terms of examples. Also, this will make clear what large compared with n/τ means. Errors introduced in the scaling and squaring portion of the computation of the table in Fig. 3.9.2 are propagated during the computation of the remaining differences. We bound these errors relative to the table in Fig. 3.9.3.

From (3.9.1-2), depending on our arithmetic assumptions,

$$|fl(\Delta_0^k\exp_\tau) - \Delta_0^k\exp_\tau| \leqslant \epsilon\kappa\cdot\Delta_{\xi_0}^k\exp_\tau$$

for each $k=0,1,\ldots,n$. To keep the analysis simple we forget that all zeroth, and even first, order differences may be computed specially with a smaller error coefficient than κ. The difference $\Delta_0^0\exp_\tau \equiv e^{\tau\zeta_0}$ in Fig. 3.9.2 is computed with an absolute error $\delta_0^0 \equiv fl(e^{\tau\zeta_0}) - e^{\tau\zeta_0}$ such that $|\delta_0^0| \leqslant \epsilon\kappa e^{\tau\xi_0}$. Now,

$$\Delta_{-0}^1\exp_\tau = \frac{1}{\eta_0}\mathrm{Im}(\Delta_0^0\exp_\tau);$$

the propagated error δ_{-0}^1 satisfies

$$|\delta_{-0}^1| \leqslant \epsilon\kappa\frac{e^{\tau\xi_0}}{\eta_0}.$$

Since $\Delta_0^1\exp_\tau$ is computed by scaling and squaring, its absolute error δ_0^1 satisfies $|\delta_0^1| \leqslant \epsilon\kappa\cdot\Delta_{\xi_0}^1\exp_\tau$. By (2.4.3) the propagated absolute error in the computed $\Delta_{-0}^2\exp_\tau$ is

$$\delta_{-0}^2 = \frac{\delta_0^1 - \delta_{-0}^1}{\zeta_1 - \zeta_{-0}},$$

thus

$$|\delta_{-0}^2| \leqslant \epsilon\kappa\frac{\Delta_{\xi_0}^1\exp_\tau + \frac{1}{\eta_0}e^{\tau\xi_0}}{|\zeta_1 - \zeta_{-0}|} \leqslant \frac{\eta_0 + 1/\tau}{|\zeta_1 - \zeta_{-0}|}\cdot\epsilon\frac{\kappa}{\eta_0}\Delta_{\xi_0}^1\exp_\tau$$

where bound (3.2.3) is used on $e^{\tau\xi_0}$. When $\eta_0 \approx \eta_1$ and η_0 is large compared with $1/\tau$,

$$\frac{\eta_0 + 1/\tau}{|\zeta_1 - \zeta_{-0}|} \approx \frac{1}{2},$$

or is even smaller than ½ when the difference in the real parts $\xi_1 - \xi_{-0}$ is large. δ_{-0}^2 satisfies the simple bound

$$|\delta_{-0}^2| \equiv |fl(\Delta_{-0}^2\exp_\tau) - \Delta_{-0}^2\exp_\tau| \leqslant \epsilon\frac{\kappa}{\eta_0}\Delta_{\xi_0}^1\exp_\tau.$$

One more step makes the general case (3.9.5) clear. Since $|\delta_0^2| \leqslant \epsilon\kappa\cdot\Delta_{\xi_0}^2\exp_\tau$ by scaling and squaring,

$$|\delta_{-0}^3| \leqslant \epsilon\kappa\frac{\Delta_{\xi_0}^2\exp_\tau + \frac{1}{\eta_0}\Delta_{\xi_0}^1\exp_\tau}{|\zeta_2 - \zeta_{-0}|}$$

$$\leqslant \frac{\eta_0 + 2/\tau}{|\zeta_2 - \zeta_{-0}|}\cdot\epsilon\frac{\kappa}{\eta_0}\Delta_{\xi_0}^2\exp_\tau.$$

Thus when $\eta_0 \approx \eta_1 \approx \eta_2$ and η_0 is large compared with $2/\tau$,

$$\frac{\eta_0 + 2/\tau}{|\zeta_2 - \zeta_{-0}|} \approx \frac{1}{2},$$

and again we have the simple bound

$$|\delta_{-0}^3| \equiv |fl(\Delta_{-0}^3\exp_\tau) - \Delta_{-0}^3\exp_\tau| \leqslant \epsilon\frac{\kappa}{\eta_0}\Delta_{\xi_0}^2\exp_\tau.$$

By continuing this process, (3.9.5) can be checked.

When the η_j are large compared with n/τ, the coefficient of 1/2 that appears above suggests the assymptotic estimates for the divided differences, as in Fig. 3.9.4.

Estimated error bounds: When each η_j, $j=0,1,\ldots,n$, is large compared with n/τ,

$$|fl(\Delta_{-k}^{2k+1}\exp_\tau) - \Delta_{-k}^{2k+1}\exp_\tau| \sim \epsilon\kappa 2^{-k}\Big(\prod_{j=0}^{k}\eta_j\Big)^{-1}\cdot\Delta_{\xi_0}^{k}\exp_\tau \tag{3.9.6a}$$

and

$$|fl(\Delta_{-k}^{2k+l}\exp_\tau) - \Delta_{-k}^{2k+l}\exp_\tau| \sim \epsilon\kappa 2^{-(k+1)}\Big(\prod_{j=0}^{k}\eta_j\Big)^{-1}\cdot\Delta_{\xi_0}^{k+l-1}\exp_\tau \tag{3.9.6b}$$

for each $k=0,1,\ldots,n$ and $l=2,\ldots,n-k+1$.

(3.9.5) shows that $\epsilon\kappa$ times a matrix like that in Fig. 3.9.3 bounds the error. (3.9.6a-b) indicates that $\epsilon\kappa$ times a matrix as in Fig. 3.9.4 is a good approximate bound. The elements in Fig. 3.9.4 are the estimated values for the conjugate differences (3.7.14). These are good estimates when the η_j are large; hence, a bound using them is nearly a relative error bound. Because κ depends only on the scaling and squaring, the standard formula portion of the hybrid algorithm does not lead to error growth, which is the purpose of reordering the data points.

example: The data from the previous example, namely

$$\begin{aligned}
\zeta_{\pm 0} &= -1.414214 \pm i8.585786\\
\zeta_{\pm 1} &= \ 1.412799 \pm i11.41563\\
\zeta_{\pm 2} &= \ 1.414214 \pm i11.41421\\
\zeta_{\pm 3} &= \ 1.417039 \pm i11.41138\,.
\end{aligned}$$

generate the divided differences shown in Fig. 3.9.6 (only half the table is exhibited). The data were generated by assigning each $\zeta_l = \alpha + \rho e^{\phi_l i}$, $l=0,1,2,3$, and rounding to seven digits. $\alpha = 10i$, $\rho = 2$, and $\phi_0 = -3\pi/4$, $\phi_1 = \pi/4 + 0.001$, $\phi_2 = \pi/4$ and $\phi_3 = \pi/4 - 0.002$. This yields both closely clustered and moderately separated data points. The arithmetic is seven digit single precision, so condition (3.4.14) with spread $\theta = 4$ gives $j = 2$ squarings. From (3.9.2) the error coefficient is

$$\kappa \equiv (3+1)[\kappa_1\cdot 4 - 1] \approx 337\,.$$

In Fig. 3.9.5 the estimated bounds are very close to the true absolute values of the

divided differences, so κ indicates a loss of, at most, 2.5 decimal digits. This κ is clearly excessive. Indeed, the double precision coefficient

$$\kappa \equiv \kappa_2 \cdot 4 - 1 \approx 32.3,$$

giving a loss of about 1.5 digits, is also larger than the results in Fig. 3.9.6 warrant.

Differences correct to seven decimal digits	
Row Index	Divided difference table
-1	(1.063108E-3, 0.0)
-0	(2.106638E-2, 0.0) (-5.269139E-2, 1.213605E-2)
0	(-1.624537E-1, 1.808715E-1) (-3.706310E-1,-1.019594)
1	(1.675059 ,-3.750361)
-3	(1.611337E-6, 0.0)
-2	(1.671230E-4, 0.0) (3.230809E-5, 1.838758E-5)
-1	(-1.114105E-4, 1.907577E-3) (-2.524930E-4, 7.375033E-4)
-0	(-4.248672E-2,-2.540785E-3) (-1.694748E-2,-3.852958E-3)
0	(-1.220463E-1,-8.447846E-1) (-1.342108E-2,-3.523510E-1)
1	(1.673579 ,-3.754205) (8.355560E-1,-1.880302)
2	(1.672096 ,-3.758050) (1.669130 ,-3.765728)
3	(1.666154 ,-3.773412)

Differences computed by hybrid algorithm	
Row Index	Divided difference table
-1	(1.063106E-3, 0.0)
-0	(2.106639E-2, 0.0) (-5.269139E-2, 1.213603E-2)
0	(-1.624537E-1, 1.808715E-1) (-3.706307E-1,-1.019594)
1	(1.675059 ,-3.750361)
-3	(1.611334E-6, 0.0)
-2	(1.671228E-4, 0.0) (3.230808E-5, 1.838754E-5)
-1	(-1.114093E-4, 1.907575E-3) (-2.524923E-4, 7.375030E-4)
-0	(-4.248669E-2,-2.540757E-3) (-1.694747E-2,-3.852943E-3)
0	(-1.220468E-1,-8.447842E-1) (-1.342132E-2,-3.523509E-1)
1	(1.673576 ,-3.754205) (8.355546E-1,-1.880302)
2	(1.672096 ,-3.758050) (1.669127 ,-3.765729)
3	(1.666154 ,-3.773412)

Fig. 3.9.6: Conjugate pair exponential divided differences.

Exponential divided differences with real abscissae are accurately computed by a hybrid type algorithm. This idea of decomposing the divided difference table into blocks, each best computed by a particular method, may be extended to additional cases, such as conjugate pair exponential divided differences. Indeed any sequence of abscissae which readily decomposes into well-separated clusters is well suited to the hybrid approach; and the idea need not be restricted to exponential divided differences. Though scaling and squaring does not work in general, the function may possess special properties which are exploitable through representing its divided difference table as a matrix function. The series algorithms are still applicable for clustered abscissae. Certainly many extensions are possible, only the simplest and most basic have been dealt with here.

Our original intention in studying divided differences was to find a quick and accurate way to compute the matrix exponential. We have always kept in mind the Newton polynomial representation and techniques appropriate for computing matrix functions. The techniques we have employed, scaling and squaring, the standard divided difference recurrence, Taylor series, and decomposing the table to apply a hybrid algorithm, all have analogues appropriate for computing the exponential of a matrix [Moler and Van Loan, 1978]. Indeed, it was these analogues that suggested many of the approaches pursued here. Thus our study of divided differences not only aids in computing more general matrix functions (the Newton polynomial), but it also provides an indication of difficulties that lie in wait in matrix function evaluations. Precisely because divided difference tables are matrix functions, a full understanding of methods for computing such tables is essential to an understanding of functions of a matrix.

Appendix

A.1 The Newton divided difference series.

For those readers who may be unfamiliar with divided difference expansions such as

$$f(\zeta) = \sum_{k=0}^{\infty} \Delta_0^k f \cdot \prod_{j=0}^{k-1} (\zeta - \alpha_j),$$

we present here a convergence proof sufficient for our purposes. Similar expansions are studied, for example, by Gel'fond [1971], but they are not quite what we require.

A simple derivation of the Newton divided difference series is obtained from the contour integral formula (2.1.13)

$$\Delta_0^n f = \frac{1}{2\pi i} \int_C \frac{f(\omega)\, d\omega}{(\omega - \alpha_0)(\omega - \alpha_1) \cdots (\omega - \alpha_n)}.$$

Our proof follows a method commonly employed to establish the convergence of complex Taylor series. The Taylor expansion of f, of course, is a special case of the more general Newton expansion.

We begin by deriving a Newton formula with remainder. The expansion points are the abscissae of the divided differences which are coefficients in the expansion.

Newton divided difference expansion with remainder. Let $\mathbf{A}_n \equiv \{\alpha_0, \alpha_1, \ldots, \alpha_n\}$ be a sequence of expansion points and let f be holomorphic on a simply connected region D containing $\mathbf{A}_n$. Then for any simple closed contour C in D enclosing $\mathbf{A}_n$ and a point ζ,

$$f(\zeta) = \sum_{k=0}^{n} \Delta_0^k f \cdot \prod_{j=0}^{k-1} (\zeta - \alpha_j) + R_n(\zeta)$$

where the remainder

$$R_n(\zeta) = \frac{1}{2\pi i} \int_C (\omega - \zeta)^{-1} f(\omega) \prod_{j=0}^{n} [(\zeta - \alpha_j)(\omega - \alpha_j)^{-1}]\, d\omega.$$

proof: From (2.7.5) where $p_n(\zeta) \equiv \prod_{j=0}^{n-1} (\zeta - \alpha_j)$,

$$\frac{p_{n+1}(\omega) - p_{n+1}(\zeta)}{\omega - \zeta} = \frac{1}{\omega - \zeta} \{\prod_{j=0}^{n} (\omega - \alpha_j) - \prod_{j=0}^{n} (\zeta - \alpha_j)\}$$

$$= \sum_{k=0}^{n} \{\prod_{j=0}^{k-1}(\zeta-\alpha_j)\cdot \prod_{l=k+1}^{n}(\omega-\alpha_l)\}.$$

Dividing by $p_{n+1}(\omega)$ and rearranging yields

$$\frac{1}{\omega-\zeta} = \sum_{k=0}^{n}\{\prod_{j=0}^{k-1}(\zeta-\alpha_j)\cdot\prod_{l=0}^{k}(\omega-\alpha_l)^{-1}\} + \frac{1}{\omega-\zeta}\prod_{j=0}^{n}[(\zeta-\alpha_j)(\omega-\alpha_j)^{-1}].$$

By Cauchy's integral formula,

$$f(\zeta) = \frac{1}{2\pi i}\int_C (\omega-\zeta)^{-1} f(\omega)\, d\omega$$

$$= \sum_{k=0}^{n}\{\frac{1}{2\pi i}\int_C f(\omega)\prod_{l=0}^{k}(\omega-\alpha_l)^{-1}\, d\omega\}\prod_{j=0}^{k-1}(\zeta-\alpha_j) + R_n(\zeta)$$

$$= \sum_{k=0}^{n}\Delta_0^k f\cdot\prod_{j=0}^{k-1}(\zeta-\alpha_j) + R_n(\zeta). \quad \square$$

When $\mathbf{A}_n$ consists of the eigenvalues of a matrix A,

$$R_n(\zeta) = \frac{1}{2\pi i}\int_C \frac{f(\omega)\, d\omega}{(\omega-\zeta)(\omega-\alpha_0)\cdots(\omega-\alpha_n)}\cdot\chi_A(\zeta).$$

$\chi_A(\zeta) = \prod_{j=0}^{n}(\zeta-\alpha_j)$ is the characteristic polynomial of A. When f is holomorphic inside and on C, the integral is bounded in ζ. $R_n(A) = 0$ by the Cayley-Hamilton theorem, thus establishing the Newton polynomial representation of $f(A)$ for holomorphic f.

We need only show that $R_n(\zeta) \to 0$ as $n \to \infty$ to establish the Newton series formula.

Newton divided difference expansion. Let $\mathbf{A} \equiv \{\alpha_0, \alpha_1, \alpha_2, \ldots\}$ be a sequence of expansion points and suppose only finitely many points of A lie outside a circle of radius ϵ about a point α. Suppose further that f is holomorphic on a simply connected region D containing A and a disk about α of radius $\rho > 2\epsilon$. Then for all ζ such that $|\zeta-\alpha| < \rho - 2\epsilon$,

$$f(\zeta) = \sum_{k=0}^{\infty}\Delta_0^k f\cdot\prod_{j=0}^{k-1}(\zeta-\alpha_j).$$

proof: Select a simple closed contour C in D enclosing A and such that

$$\rho_C \equiv \min_{\omega\in C}|\omega-\alpha| > |\zeta-\alpha| + 2\epsilon.$$

Because f is holomorphic on C, there exists a constant K such that $|f(\omega)| \leqslant K$ for all $\omega \in C$. Let $M \equiv \{j \mid \epsilon < |\alpha_j - \alpha|,\ j=0,1,2,\ldots\}$ and let m be the number of elements in M. Since m is finite,

$$\beta(\zeta) \equiv \max\{1, \max_{j \in M} \max_{\omega \in C'} \frac{|\zeta - \alpha_j|}{|\omega - \alpha_j|}\}$$

exists. C was selected so that $\omega \neq \alpha_j$ for any j. C was also selected such that for all $\omega \in C$

$$|\omega - \zeta| > |\omega - \alpha| - |\zeta - \alpha| \geqslant 2\epsilon,$$

and

$$\frac{|\zeta - \alpha_j|}{|\omega - \alpha_j|} \leqslant \frac{|\zeta - \alpha| + \epsilon}{\rho_{C'} - \epsilon} \equiv \gamma(\zeta) < 1$$

for all $j \in M^c$, the complement of M. Thus

$$|R_n(\zeta)| \leqslant \frac{1}{2\pi} \int_C \frac{|f(\omega)|}{|\omega - \zeta|} \cdot \prod_{\substack{j \in M \\ j \leqslant n}} \frac{|\zeta - \alpha_j|}{|\omega - \alpha_j|} \cdot \prod_{\substack{j \in M^c \\ j \leqslant n}} \frac{|\zeta - \alpha_j|}{|\omega - \alpha_j|} \cdot |d\omega|$$

$$\leqslant \frac{1}{2\pi} \cdot \frac{K}{2\epsilon} [\beta(\zeta)]^m [\gamma(\zeta)]^{n-m} L$$

where L is the length of C. Then as $n \to \infty$, $|R_n(\zeta)| \to 0$. □

On every closed disk $|\zeta - \alpha| \leqslant \rho'$ where $\rho' < \rho - 2\epsilon$, the series converges uniformly to f. ρ may be chosen as the radius of the largest open disk about α in D. When the sequence of expansion points $\{\alpha_0, \alpha_1, \alpha_2, \ldots\}$ converges to α, the ϵ of the theorem may be chosen arbitrarily small. Convergence of the Newton expansion may then be claimed for all ζ such that $|\zeta - \alpha| < \rho$. In particular when all the expansion points are confluent at α, the Taylor expansion of f appears as a corollary.

Taylor expansion. Suppose f is holomorphic on a simply connected region **D** that contains a disk of radius ρ about α. Then for all ζ such that $|\zeta - \alpha| < \rho$,

$$f(\zeta) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\alpha)}{k!}(\zeta - \alpha)^k.$$

proof: Recall that $\Delta_0^k f = f^{(k)}(\alpha)/k!$ for confluent abscissae. □

It should also be noted that because f is holomorphic on **D**, the theorem applies equally well to any derivative of f.

A.2 Divided difference expansion of matrix functions.

The results of the previous section will now be extended to functions of a matrix. For a $(n+1)\times(n+1)$ matrix A, the matrix function $f(A)$ has a Newton series representation when A's eigenvalues all lie inside the series' circle of convergence.

Newton divided difference series for a matrix. Suppose f has a Newton series expansion (as in §A.1) on the disk $\mathbf{D}_\rho \equiv \{\zeta \mid \rho - 2\epsilon > |\zeta - \alpha|\}$. Then if every eigenvalue λ_i, $0 \leqslant i \leqslant n$, of A lies in $\mathbf{D}_\rho$,

$$f(A) = \sum_{k=0}^{\infty} \Delta_0^k f \cdot \prod_{j=0}^{k-1} (A - \alpha_j I) .$$

proof: For any $\omega \neq \lambda_i$, $0 \leqslant i \leqslant n$, the matrix $(\omega I - A)$ is non-singular and

$$(\omega I - A)^{-1} = \sum_{k=0}^{n} \{\prod_{j=0}^{k-1} (A - \alpha_j I) \cdot \prod_{j=0}^{k} (\omega - \alpha_j)^{-1}\} + \prod_{j=0}^{n} \frac{A - \alpha_j I}{\omega - \alpha_j} (\omega I - A)^{-1} .$$

By the Cartan definition (1.1.7)

$$f(A) = \frac{1}{2\pi i} \int_C f(\omega)(\omega I - A)^{-1} d\omega .$$

The simple closed contour C is selected such that it encloses all the expansion points and

$$\rho_C \equiv \min_{\omega \in C} |\omega - \alpha| > \max_{0 \leqslant i \leqslant n} |\lambda_i - \alpha| + 2\epsilon .$$

Then

$$f(A) = \sum_{k=0}^{n} \Delta_0^k f \cdot \prod_{j=0}^{k-1} (A - \alpha_j I) + R_n(A)$$

where the remainder

$$R_n(A) = \frac{1}{2\pi i} \int_C f(\omega)(\omega I - A)^{-1} \prod_{j=0}^{n} \frac{A - \alpha_j I}{\omega - \alpha_j} d\omega .$$

To complete the proof, we need only show that in some norm $\|R_n(A)\| \to 0$ as $n \to \infty$.

Define the set M as in the proof in §A.1. Then†

† $\|B\|_\infty = \max_{0 \leqslant i \leqslant n} \{\sum_{j=0}^{n} |b_{ij}|\}$.

$$\|R_n(A)\|_\infty \leqslant \frac{1}{2\pi}\int_C |f(\omega)|\cdot\|(\omega I - A)^{-1}\|_\infty \cdot \prod_{\substack{j\in M\\ j\leqslant n}} \|\frac{A-\alpha_j I}{\omega-\alpha_j}\|_\infty \cdot \|\prod_{\substack{j\in M^c\\ j\leqslant n}} \frac{A-\alpha_j}{\omega-\alpha_j}\|_\infty \cdot |d\omega|\,.$$

The curve C has finite length L. For all $\omega \in C$,

$$|f(\omega)| \leqslant K$$

for some constant K because f is holomorphic on C, and

$$\|(\omega I - A)^{-1}\|_\infty \leqslant K'$$

for some other constant K' because C is bounded away from A's eigenvalues. The constant

$$\beta \equiv \max\{1, \max_{j\in M}\max_{\omega\in C} \|\frac{A-\alpha_j I}{\omega-\alpha_j}\|_\infty\}$$

exists because M is a finite set and C does not contain any α_j. For all $j \in M^c$ each eigenvalue λ_i of A, $0 \leqslant i \leqslant n$, satisfies the inequalities

$$\max_{\omega\in C} \frac{|\lambda_i-\alpha_j|}{|\omega-\alpha_j|} \leqslant \frac{|\lambda_i-\alpha|+\epsilon}{\rho_C-\epsilon} \leqslant \max_{0\leqslant i\leqslant n} \frac{|\lambda_i-\alpha|+\epsilon}{\rho_C-\epsilon} \equiv \gamma' < 1\,.$$

Let $A = P^{-1}JP$ where the upper triangular matrix J is A's Jordan canonical form. Then

$$\prod_{\substack{j\in M^c\\ j\leqslant n}} \frac{A-\alpha_j I}{\omega-\alpha_j} = P^{-1}\cdot \prod_{\substack{j\in M^c\\ j\leqslant n}} \frac{J-\alpha_j I}{\omega-\alpha_j}\cdot P$$

$$= P^{-1}D\cdot \prod_{\substack{j\in M^c\\ j\leqslant n}} \frac{D^{-1}JD-\alpha_j I}{\omega-\alpha_j}\cdot D^{-1}P$$

where $D \equiv \mathrm{diag}(1, \eta, \eta^2, \ldots, \eta^n)$, $\eta > 0$. Taking norms,

$$\|\prod_{\substack{j\in M^c\\ j\leqslant n}} \frac{A-\alpha_j I}{\omega-\alpha_j}\|_\infty \leqslant \|P^{-1}D\|_\infty\cdot\|D^{-1}P\|_\infty\cdot \prod_{\substack{j\in M^c\\ j\leqslant n}} \|\frac{D^{-1}JD-\alpha_j I}{\omega-\alpha_j}\|_\infty\,.$$

For any $j \in M^c$

$$\|\frac{D^{-1}JD-\alpha_j I}{\omega-\alpha_j}\|_\infty \leqslant \max_{0\leqslant i\leqslant n} \frac{|\lambda_i-\alpha_j|}{|\omega-\alpha_j|} + \frac{\eta}{|\omega-\alpha_j|}$$

$$\leqslant \gamma' + \frac{\eta}{\rho_C-\epsilon} \equiv \gamma < 1$$

for all $\eta < (\rho_C - \epsilon)(1-\gamma')$. Thus

$$\Big\| \prod_{\substack{j \in M^c \\ j \le n}} \frac{A - \alpha_j I}{\omega - \alpha_j} \Big\|_\infty \le K'' \gamma^{n-m},$$

where the constant $K'' \equiv \|P^{-1}D\|_\infty \cdot \|D^{-1}P\|_\infty$ for some fixed $\eta < (\rho_C - \epsilon)(1-\gamma')$. Combining these bounds yields

$$\|R_n(A)\|_\infty \le \frac{L}{2\pi} K K' \beta^m K'' \gamma^{n-m},$$

and $\|R_n(A)\|_\infty \to 0$ as $n \to \infty$. □

B. Error bounds for a Taylor series computation of $\Delta_0^n \exp_\tau$.

The Taylor series is one of a number of proposed methods for computing $\Delta_0^n \exp_\tau$. It involves no previously computed divided differences; so its study does not concern propagation of errors, but just computational errors. Here we develop error bounds on the computation of $\Delta_0^n \exp_\tau$ by a Taylor series and demonstrate that the method is best applied when the abscissae are closely clustered.

In §2.8 $\Delta_0^n \exp_\tau$, with real or complex abscissae $\{\zeta_0, \zeta_1, \ldots, \zeta_n\}$ and $\tau \geqslant 0$, is shown to have a Taylor expansion about α

$$\Delta_0^n \exp_\tau = \sum_{j=0}^{\infty} \frac{\tau^{n+j} e^{\tau\alpha}}{(n+j)!} \Delta_0^n \uparrow_\alpha^{n+j}. \tag{B.1}$$

where the power function $\uparrow_\alpha^{n+j}$ is $\uparrow_\alpha^{n+j}(\zeta) = (\zeta-\alpha)^{n+j}$, $j = 0, 1, 2, \ldots$. It is convenient to consider the shifted abscissae $\{\zeta_0-\alpha, \zeta_1-\alpha, \ldots, \zeta_n-\alpha\}$ exact; the numerical effects of shifting abscissae are discussed in §3.6. With

$$\delta \equiv \max_{0 \leqslant i \leqslant n} |\zeta_i - \alpha|, \tag{B.2}$$

the bounds we obtain resemble

$$|fl(\Delta_0^n \exp_\tau) - \Delta_0^n \exp_\tau| \leqslant \mu e^{\tau\delta} \frac{\tau^n |e^{\tau\alpha}|}{n!},$$

where μ represents a coefficient dependent on the arithmetic details to be introduced shortly. $fl(\Delta_0^n \exp_\tau)$ represents the computed floating-point value of $\Delta_0^n \exp_\tau$.

The Taylor series algorithm outlined in §3.3 requires many inner products. We consider two separate conditions for bounding round-off error accumulation in inner product computations.†

1. Double precision accumulation. The error in the computed inner product $fl_2(\sum_{i=0}^{n} \alpha_i \beta_i)$ satisfies

$$|fl_2(\sum_{i=0}^{n} \alpha_i \beta_i) - \sum_{i=0}^{n} \alpha_i \beta_i| \leqslant \epsilon \sum_{i=0}^{n} |\alpha_i \beta_i|, \tag{B.3}$$

†See Wilkinson [1963] for a general treatment of rounding error analysis.

where $\epsilon \equiv 1.06 \times$ (machine precision). ϵ is assumed so small that any $O(\epsilon^2)$ expressions are negligible when compared with expressions linear in ϵ and may, because of the arbitrary 1.06, be absorbed into such linear expressions. Error condition (B.3) holds for double precision accumulation of sums and inner products. It does not depend on the number of terms summed and leads to simple illustrative error bounds. Additionally, we assume the series coefficients $\tau^{n+j}e^{\tau\alpha}/(n+j)!$ are all calculable to machine precision, namely

$$|fl_2(\frac{\tau^{n+j}e^{\tau\alpha}}{(n+j)!}) - \frac{\tau^{n+j}e^{\tau\alpha}}{(n+j)!}| < \epsilon\frac{\tau^{n+j}|e^{\tau\alpha}|}{(n+j)!}\,. \tag{B.4}$$

2. Single precision accumulation. The second condition applies to single precision computation of all quantities. Wilkinson [1963] shows that

$$|fl(\sum_{i=0}^{n}\alpha_i\beta_i) - \sum_{i=0}^{n}\alpha_i\beta_i| \leqslant \epsilon[(n+1)|\alpha_0\beta_0| + \sum_{i=1}^{n}(n+2-i)|\alpha_i\beta_i|]\,.$$

We simplify this to the more convenient

$$|fl(\sum_{i=0}^{n}\alpha_i\beta_i) - \sum_{i=0}^{n}\alpha_i\beta_i| \leqslant \epsilon\sum_{i=0}^{n}(n+2-i)|\alpha_i\beta_i|\,. \tag{B.5}$$

In addition, we assume the series coefficients are evaluable with no more than five rounding errors (say, errors in the evaluation of τ^{n+j}, $e^{\tau\alpha}$, and $(n+j)!$, plus a multiplication and a division); so

. (B.6)

Bounds derived from (B.5), though more complex, are more generally applicable than those from the first condition.

We start by deriving bounds on divided differences of power functions.

Lemma 1: For $j = 1, 2, \ldots$

$$|\Delta_0^k\uparrow_\alpha^{j+k}| \leqslant \frac{(j+k)!\,\delta^j}{k!\,j!}\,, \quad k = 0, 1, \ldots, n\,. \tag{B.7}$$

proof: From the recurrence (2.7.8),

$$\Delta_0^k\uparrow_\alpha^{j+k} = (\zeta_k - \alpha)\cdot\Delta_0^k\uparrow_\alpha^{j+k-1} + \Delta_0^{k-1}\uparrow_\alpha^{j+k-1}$$

$$= (\zeta_k - \alpha)\cdot\Delta_0^k\uparrow_a^{j+k-1} + (\zeta_{k-1} - \alpha)\cdot\Delta_0^{k-1}\uparrow_a^{j+k-2} + \Delta_0^{k-2}\uparrow_a^{j+k-2}$$

$$= \cdots = \sum_{i=0}^{k}(\zeta_i - \alpha)\cdot\Delta_0^i\uparrow_a^{j+i-1}.$$

For $j = 1$, $\Delta_0^k\uparrow_a^{k+1} = \sum_{i=0}^{k}(\zeta_i - \alpha)$ and so

$$|\Delta_0^k\uparrow_a^{k+1}| \leqslant (k+1)\delta, \quad k = 0, 1, \ldots, n.$$

If for some $j \geqslant 1$

$$|\Delta_0^k\uparrow_a^{j+k-1}| \leqslant \frac{(j+k-1)!}{k!\,(j-1)!}\delta^{j-1}$$

for each $k = 0, 1, \ldots, n$, then since $\Delta_0^k\uparrow_a^{j+k} = \sum_{i=0}^{k}(\zeta_i - \alpha)\cdot\Delta_0^i\uparrow_a^{j+i-1}$,

$$|\Delta_0^k\uparrow_a^{j+k}| \leqslant \frac{\delta^j}{(j-1)!}\sum_{i=0}^{k}\frac{(j+i-1)!}{i!} = \frac{(j+k)!}{k!\,j!}\delta^j$$

for $k = 0, 1, \ldots, n$. □

We now give bounds on the error in $fl(\Delta_0^k\uparrow_a^{j+k})$ for each k and j. When the error is not too large, $fl(\Delta_0^k\uparrow_a^{j+k})$ may also be bounded as in Lemma 1.

Lemma 2: Let $\Delta_0^k\uparrow_a^{j+k}$ be computed according to Algorithm 1 of §2.7. Then for each $j = 1, 2, \ldots$ and $k = 0, 1, \ldots, n$,

$$|fl_2(\Delta_0^k\uparrow_a^{j+k}) - \Delta_0^k\uparrow_a^{j+k}| \leqslant \epsilon\delta^j\frac{(j+k)!}{k!\,(j-1)!} = j\epsilon\delta^j\binom{j+k}{k} \tag{B.8a}$$

for double precision accumulation (B.3), while for single precision accumulation (B.5),

$$|fl(\Delta_0^k\uparrow_a^{j+k}) - \Delta_0^k\uparrow_a^{j+k}| \leqslant \epsilon\delta^j\left[j\frac{(j+k+1)!}{k!\,(j+1)!} + (j-1)\frac{(j+k)!}{k!\,j!}\right]. \tag{B.8b}$$

proof: For (B.8a),

$$|fl_2(\Delta_0^k\uparrow_a^{k+1}) - \Delta_0^k\uparrow_a^{k+1}| \leqslant \epsilon\sum_{i=0}^{k}|\zeta_i - \alpha| \leqslant (k+1)\delta\epsilon, \quad k = 0, 1, \ldots, n,$$

by (B.3) when $j = 1$. If for some $j \geqslant 1$

$$|\mathcal{H}_2(\mathbb{A}_0^k\uparrow_a^{j+k-1}) - \mathbb{A}_0^k\uparrow_a^{j+k-1}| \leq \epsilon\delta^{j-1}\frac{(j+k-1)!}{k!\,(j-2)!}$$

for each $k = 0, 1, \ldots, n$, then

$$|\mathcal{H}_2(\mathbb{A}_0^k\uparrow_a^{j+k}) - \mathbb{A}_0^k\uparrow_a^{j+k}| \leq |\mathcal{H}_2[\sum_{i=0}^{k}(\zeta_i - \alpha)\mathcal{H}_2(\mathbb{A}_0\uparrow_a^{j+i-1})] - \sum_{i=0}^{k}(\zeta_i - \alpha)\mathcal{H}_2(\mathbb{A}_0\uparrow_a^{j+i-1})|$$

$$+ \sum_{i=0}^{k}|\zeta_i - \alpha|\,|\mathcal{H}_2(\mathbb{A}_0\uparrow_a^{j+i-1}) - \mathbb{A}_0\uparrow_a^{j+i-1}|$$

$$\leq \epsilon\delta^j[\frac{1}{(j-1)!} + \frac{1}{(j-2)!}]\sum_{i=0}^{k}\frac{(j+i-1)!}{i!}$$

$$= \epsilon\delta^j\frac{(j+k)!}{k!\,(j-1)!}\,.$$

Similarly for (B.8b), when $j = 1$,

$$|\mathcal{H}(\mathbb{A}_0^k\uparrow_a^{k+1}) - \mathbb{A}_0^k\uparrow_a^{k+1}| \leq \epsilon\sum_{i=0}^{k}(k+1-i)\,|\zeta_i - \alpha|$$

$$\leq \epsilon\delta\frac{(k+2)(k+1)}{2!}\,, \quad k = 0, 1, \ldots, n\,,$$

by (B.5) modified slightly to reflect that $\mathbb{A}_0^k\uparrow_a^{k+1}$ is just a sum. Now if for some $j \geq 1$

$$|\mathcal{H}(\mathbb{A}_0^k\uparrow_a^{j+k-1}) - \mathbb{A}_0^k\uparrow_a^{j+k-1}| \leq \epsilon\delta^{j-1}[(j-1)\frac{(j+k)!}{k!\,j!} + (j-2)\frac{(j+k-1)!}{k!\,(j-1)!}]$$

for each $k = 0, 1, \ldots, n$, then

$$|\mathcal{H}(\mathbb{A}_0^k\uparrow_a^{j+k}) - \mathbb{A}_0^k\uparrow_a^{j+k}| \leq \epsilon\sum_{i=0}^{k}(k+2-i)\,|\zeta_i - \alpha|\,|\mathcal{H}(\mathbb{A}_0\uparrow_a^{j+i-1})|$$

$$+ \sum_{i=0}^{k}|\zeta_i - \alpha|\,|\mathcal{H}(\mathbb{A}_0\uparrow_a^{j+i-1}) - \mathbb{A}_0\uparrow_a^{j+i-1}|$$

$$\leq \epsilon\delta^j\sum_{i=0}^{k}[(k+2-i)\frac{(j+i-1)!}{i!\,(j-1)!} + (j-1)\frac{(j+i)!}{i!\,j!} + (j-2)\frac{(j+i-1)!}{i!\,(j-1)!}]$$

$$= \epsilon\delta^j[\frac{(j+k+1)!}{k!\,(j+1)!} + (j-1)\frac{(j+k)!}{k!\,j!}]\,. \quad \square$$

Theorem: In computing $\Delta_0''\exp_\tau$ by a Taylor expansion about α, the error is bounded by

$$|fl_2(\Delta_0''\exp_\tau) - \Delta_0''\exp_\tau| \leqslant \epsilon e^{\tau\delta}(2+\tau\delta)\frac{\tau^n|e^{\tau\alpha}|}{n!} \tag{B.9a}$$

for double precision accumulation (B.3), while for single precision accumulation (B.5)

$$|fl(\Delta_0''\exp_\tau) - \Delta_0''\exp_\tau| \leqslant \epsilon(m+n+7+\tau\delta)e^{\tau\delta}\frac{\tau^n|e^{\tau\alpha}|}{n!}, \tag{B.9b}$$

where $m+1$ is the number of terms actually summed in the expansion.

proof: Let $\Delta_0''s_{n+m} \equiv \sum_{j=0}^{m}\beta_{n+j}\cdot\Delta_0''\uparrow_\alpha^{n+j}$ be partial sums of the Taylor expansion (B.1), where each coefficient $\beta_{n+j} \equiv \tau^{n+j}e^{\tau\alpha}/(n+j)!$. The error is bounded by four terms,

$$\begin{aligned}
|fl(\Delta_0''\exp_\tau) - \Delta_0''\exp_\tau| \leqslant\ & |fl(\Delta_0''\exp_\tau) - \sum_{j=0}^{m} fl(\beta_{n+j})\,fl(\Delta_0''\uparrow_\alpha^{n+j})| \\
& + |\sum_{j=0}^{m} fl(\beta_{n+j})\cdot fl(\Delta_0''\uparrow_\alpha^{n+j}) - \sum_{j=0}^{m}\beta_{n+j}\,fl(\Delta_0''\uparrow_\alpha^{n+j})| \\
& + |\sum_{j=0}^{m}\beta_{n+j}\cdot fl(\Delta_0''\uparrow_\alpha^{n+j}) - \sum_{j=0}^{m}\beta_{n+j}\cdot\Delta_0''\uparrow_\alpha^{n+j}| \\
& + |\Delta_0''s_{n+m} - \Delta_0''\exp_\tau| \\
\equiv\ & \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV}.
\end{aligned}$$

We bound each of the four terms separately. In addition we note that

$$fl(\Delta_0''\exp_\tau) \equiv fl[\sum_{j=0}^{m} fl(\beta_{n+j})\cdot fl(\Delta_0''\uparrow_\alpha^{n+j})].$$

Double precision accumulation (B.3): By (B.3) and (B.7),

$$\mathrm{I} \leqslant \epsilon\sum_{j=0}^{m}|fl_2(\beta_{n+j})|\,|fl_2(\Delta_0''\uparrow_\alpha^{n+j})| \leqslant \epsilon\sum_{j=0}^{m}\frac{\tau^{n+j}|e^{\tau\alpha}|}{(n+j)!}\cdot\frac{(n+j)!\,\delta^j}{n!\,j!}$$

$$= \epsilon\frac{\tau^n|e^{\tau\alpha}|}{n!}\sum_{j=0}^{m}\frac{\tau^j\delta^j}{j!} \leqslant \epsilon e^{\tau\delta}\frac{\tau^n|e^{\tau\alpha}|}{n!}.$$

By (B.4) and (B.7),

$$\mathrm{II} \leqslant \sum_{j=0}^{m} |fl_2(\beta_{n+j}) - \beta_{n+j}|\,|fl_2(\blacktriangle_0''\uparrow_\alpha''^{+j})|$$

$$\leqslant \epsilon \sum_{j=0}^{m} |\beta_{n+j}|\,|fl_2(\blacktriangle_0''\uparrow_\alpha''^{+j})| \leqslant \epsilon e^{\tau\delta} \frac{\tau^n |e^{\tau\alpha}|}{n!}.$$

And using (B.8a),

$$\mathrm{III} \leqslant \sum_{j=0}^{m} |\beta_{n+j}|\,|fl_2(\blacktriangle_0''\uparrow_\alpha''^{+j}) - \blacktriangle_0''\uparrow_\alpha''^{+j}| \leqslant \epsilon \sum_{j=1}^{m} \frac{\tau^{n+j}|e^{\tau\alpha}|}{(n+j)!} \cdot \frac{(n+j)!\,\delta^j}{n!\,(j-1)!}$$

$$= \epsilon \frac{\tau^n |e^{\tau\alpha}|}{n!} \sum_{j=1}^{m} \frac{\tau^j \delta^j}{(j-1)!} \leqslant \epsilon\tau\delta e^{\tau\delta} \frac{\tau^n |e^{\tau\alpha}|}{n!}.$$

We may ignore the truncation error IV because

$$\mathrm{IV} \leqslant |\sum_{j=0}^{m} \beta_{n+j} \cdot \blacktriangle_0''\uparrow_\alpha''^{+j} - \sum_{j=0}^{\infty} \beta_{n+j} \cdot \blacktriangle_0''\uparrow_\alpha''^{+j}|$$

$$\leqslant \sum_{j=m+1}^{\infty} |\beta_{n+j}|\,|\blacktriangle_0''\uparrow_\alpha''^{+j}| \leqslant \frac{\tau^n |e^{\tau\alpha}|}{n!} \sum_{j=m+1}^{\infty} \frac{\tau^j \delta^j}{j!}$$

is negligible for m large enough. Summing the bounds, then, yields (B.9a).

Single precision accumulation (B.5): The same steps are repeated for (B.9b). By (B.5) and (B.7),

$$\mathrm{I} \leqslant \epsilon \sum_{j=0}^{m} (m+2-j)\,|fl(\beta_{n+j})|\,|fl(\blacktriangle_0''\uparrow_\alpha''^{+j})| \leqslant \epsilon \sum_{j=0}^{m} (m+2-j) \frac{\tau^{n+j}|e^{\tau\alpha}|}{(n+j)!} \cdot \frac{(n+j)!\,\delta^j}{n!\,j!}$$

$$= \epsilon \frac{\tau^n |e^{\tau\alpha}|}{n!} \sum_{j=0}^{m} (m+2-j) \frac{\tau^j \delta^j}{j!}.$$

In the same manner as before, by (B.6)

$$\mathrm{II} \leqslant 5\epsilon e^{\tau\delta} \frac{\tau^n |e^{\tau\alpha}|}{n!}.$$

By (B.8b)

$$\mathrm{III} \leqslant \sum_{j=0}^{m} |\beta_{n+j}|\,|fl(\blacktriangle_0''\uparrow_\alpha''^{+j}) - \blacktriangle_0''\uparrow_\alpha''^{+j}|$$

$$\leqslant \epsilon \sum_{j=1}^{m} \frac{\tau^{n+j}|e^{\tau\alpha}|}{(n+j)!} \{j \frac{(j+n+1)!}{n!\,(j+1)!} + (j-1) \frac{(j+n)!}{n!\,j!}\} \delta^j$$

$$= \epsilon \frac{\tau^n |e^{\tau\alpha}|}{n!} \sum_{j=1}^{m} (2j + \frac{jn}{j+1} - 1) \frac{\tau^j \delta^j}{j!},$$

and so

$$\text{I} + \text{III} \leqslant \epsilon \frac{\tau^n |e^{\tau\alpha}|}{n!} \sum_{j=0}^{m} (m+1+\frac{jn}{j+1}+j) \frac{\tau^j \delta^j}{j!}$$

$$\leqslant \epsilon (m+n+1+\tau\delta) e^{\tau\delta} \frac{\tau^n |e^{\tau\alpha}|}{n!}.$$

Finally, we choose m so large that, say,

$$\sum_{j=m+1}^{\infty} \frac{\tau^j \delta^j}{j!} \leqslant \epsilon e^{\tau\delta}. \tag{B.10}$$

From the discussion earlier for IV,

$$\text{IV} \leqslant \epsilon e^{\tau\delta} \frac{\tau^n |e^{\tau\alpha}|}{n!}.$$

Summing our bounds on I + III, II and IV yields (B.9b). □

$\tau^n e^{\tau\alpha}/n!$ is $\Delta_0^n \exp_\tau$ for abscissae confluent at α, the bounds (B.9a-b) make clear why the Taylor series method is best applied to closely clustered abscissae.

Relation (B.10) permits determination of m when a particular ϵ and $\tau\delta$ are given. For example when $\epsilon = 10^{-7}$ and $\tau\delta \leqslant 1$, (B.10) yields

$$\sum_{j=m+1}^{\infty} \frac{1}{j!} \leqslant \epsilon e^1 \approx 2.72 \times 10^{-7}.$$

The smallest value of m for which this inequality holds is $m = 10$. And when $\epsilon = 10^{-14}$, the smallest m is $m = 16$.

C. Decision criteria for the hybrid algorithm.

Double precision accumulation. In §3.5 we found that the decision criterion $\tau\theta_n$ for the hybrid divided difference algorithm, with double precision accumulation (3.3.3), satisfies the *recurrence* (3.5.2)

$$(\kappa_2\tau\theta_n - 1) - (\kappa_2\tau\theta_{n-1} - 1)(1 + 2n/\tau\theta_n) = 0 \tag{C.1}$$

where $\tau\theta_0 \equiv 2/\kappa_2$. This recurrence has no simple closed form solution for $\tau\theta_n$. However it is possible to give a simple bound on $\tau\theta_n$.

The recurrence (C.1) is quadratic in $\tau\theta_n$; thus

$$\tau\theta_n = \frac{1}{2}\{\tau\theta_{n-1} + \sqrt{(\tau\theta_{n-1})^2 + 8n(\tau\theta_{n-1} - 1/\kappa_2)}\} \tag{C.2}$$

is a rearrangement of (C.1) where $\tau\theta_n$ appears explicitly. We attempt to bound $\tau\theta_n$ for every n by finding a function in n which satisfies a majorizing recurrence. A little exercise in completing the square gives

$$n(n+1)+2/\kappa_2 = \frac{1}{2}\{n(n-1)+2/\kappa_2 + \sqrt{[n(n-1)+2/\kappa_2]^2 + 8n[n(n-1)+2/\kappa_2 - 1/\kappa_2] + 16n^2 + 8n/\kappa_2}\},$$

which is nearly the same as (C.2). Since $\tau\theta_0 = 2/\kappa_2$, it is clear that

$$\tau\theta_n \leqslant n(n+1) + 2/\kappa_2 \tag{C.3}$$

for all $n \geqslant 0$ and any $\kappa_2 > 0$. Also in a similar way,

$$n(n-3) = \frac{1}{2}\{(n-1)(n-4) + \sqrt{[(n-1)(n-4)]^2 + 8n[(n-1)(n-4) - 1/\kappa_2] + 16n + 8n/\kappa_2}\}.$$

We compare this with recurrence (C.2). For the value of $\kappa_2 = 8.3259$ derived in §3.4, we find that $\tau\theta_{17} = 237.85 < 17\cdot(17-3) = 238$ from the table in Fig. 3.5.3. Thus $\tau\theta_n < n(n-3)$ for all $n \geqslant 17$. However,

$$n(n-4) = \frac{1}{2}\{(n-1)(n-5) + \sqrt{[(n-1)(n-5)]^2 + 8n[(n-1)(n-5) - 1/\kappa_2] - 4n^2 + 40n + 8n/\kappa_2}\}.$$

For $\kappa_2 = 8.3259$, $-4n^2 + 40n + 8n/\kappa_2 < 0$ when $n \geqslant 11$. Since $\tau\theta_{10} = 72.02 > 10\cdot(10-4) = 60$, $\tau\theta_n > n(n-4)$ for all $n \geqslant 10$. Combining these two results yields, for $n \geqslant 17$ and $\kappa_2 = 8.3259$, that $\tau\theta_n$ is bracketed by

$$n(n-4) < \tau\theta_n < n(n-3). \tag{C.4}$$

Single precision accumulation. The decision criterion $\tau\theta_n$ for single precision accumulation (3.3.4) satisfies the recurrence (3.5.6), namely

$$(n+1)(\kappa_1\tau\theta_n - 1) - n(\kappa_1\tau\theta_{n-1} - 1)(1 + 2n/\tau\theta_n) = 0. \tag{C.5}$$

This is also quadratic in $\tau\theta_n$; so we have the equivalent recurrence

$$\tau\theta_n = \frac{1}{2\kappa_1(n+1)}\{(\kappa_1 n\tau\theta_{n-1} + 1) + \sqrt{(\kappa_1 n\tau\theta_{n-1} + 1)^2 + 8\kappa_1 n^2(n+1)(\kappa_1\tau\theta_{n-1} - 1)}\} \tag{C.6}$$

in which $\tau\theta_n$ appears explicitly. Initially $\tau\theta_1 = 3/2\kappa_1$.

For $\sigma_n \equiv 2n^2/3 + (3/2\kappa_1 - 2/3)n$, we find by completing the square that

$$\sigma_n = \frac{1}{2\kappa_1(n+1)}\{(\kappa_1 n\sigma_{n-1} + 1) + \sqrt{(\kappa_1 n\sigma_{n-1} + 1)^2 + 8\kappa_1 n^2(n+1)(\kappa_1\sigma_{n-1} - 1) + \nu_n}\} \tag{C.7}$$

where $\sigma_{n-1} = 2(n-1)^2/3 + (3/2\kappa_1 - 2/3)(n-1)$ and

$$\nu_n = 4\kappa_1(n+1)\{(2 + \frac{4\kappa_1}{3})n^3 + (\frac{9}{4\kappa_1} - \frac{16\kappa_1}{9} + \frac{4}{3})n^2 + (\frac{2}{3} - \frac{3}{2\kappa_1})n\}.$$

σ_n was chosen so that $\sigma_1 = 3/2\kappa_1 = \tau\theta_1$ and $\nu_n > 0$ for any $\kappa_1 > 0$ when $n \geqslant 2$. Comparing the recurrences for σ_n and $\tau\theta_n$ shows that

$$\tau\theta_n \leqslant \frac{2}{3}n^2 + (\frac{3}{2\kappa_1} - \frac{2}{3})n \tag{C.8}$$

for $n \geqslant 1$.

To bracket $\tau\theta_n$, for large n, $\sigma_n \equiv n(2n-5)/3$ satisfies the recurrence (C.7) with

$$\nu_n = 4\kappa_1(n+1)\{(2\kappa_1 + \frac{4}{3})n^2 + \frac{5}{3}n\}.$$

A check of the first and fourth columns of Fig. 3.5.4, which has $\kappa_1 = 21.2950$, reveals that $3.16 = \tau\theta_4 < \sigma_4 = 4$. Thus $\tau\theta_n < \sigma_n \equiv n(2n-5)/3$ for all $n \geqslant 4$. Similarly, $\sigma_n \equiv 2n(n-3)/3$ satisfies the recurrence (C.7) with

$$\nu_n = 4\kappa_1(n+1)\{-\frac{4\kappa_1}{9}n^3 + (4\kappa_1 + \frac{4}{3})n^2 + 2n\}.$$

When $n \geqslant 10$, $\nu_n < 0$. From Fig. 3.5.4, $120.12 = \tau\theta_{15} > \sigma_{15} = 120$. Thus $\tau\theta_n > 2n(n-3)/3$ for $n \geqslant 15$. Combining the two bounds shows that $\tau\theta_n$ is bracketed by

$$\frac{2}{3}n^2 - 2n < \tau\theta_n < \frac{2}{3}n^2 - \frac{5}{3}n \tag{C.9}$$

for $n \geqslant 15$ and $\kappa_1 = 21.2950$.

D. Numerical examples.

The tables on the following pages illustrate the example in §3.6 of the hybrid algorithm with clustering. The first table (3 pages) is the hybrid computation in single precision for $\tau = 1$. The correct seven digit divided differences are presented in the following table for comparison. The two following tables exhibit in a digits lost ($\log_{10}$) form the actual relative error and the results of an a priori error bound computation. The data in these tables are summarized by Fig. 3.6.3. A second set of tables for $\tau = 2$ then follows (see Fig. 3.6.4). Finally for comparison, the table for $\tau = 2$ is recomputed by scaling and squaring only (Fig. 3.6.5). The abscissae are listed to the left of each table. The computations were performed on a PDP-11 computer, which has a precision slightly greater than seven decimal digits.

Divided difference table by the hybrid algorithm for tau = 1.00

abscissae	table of divided differences							
(-34.50)	.1039538e-14	.2268571e-14	.1498804e-14	.8015853e-10	.6755117e-10	.2879424e-10	.8262803e-11	.1891783e-11
(-33.10)	.0000000e+00	.4215541e-14	.4666655e-14	.1611188e-08	.1437937e-08	.6463155e-09	.1948766e-09	.4685518e-10
(-32.90)	.0000000e+00	.0000000e+00	.5148857e-14	.3012922e-07	.2850061e-07	.1352404e-07	.4290508e-08	.1085125e-08
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.5573906e-06	.5573906e-06	.2786953e-06	.9289843e-07	.2469086e-07
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5573906e-06	.5573906e-06	.2786953e-06	.1003057e-06
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5573906e-06	.5573906e-06	.3087869e-06
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5573906e-06	.6500264e-06
(-14.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.7523980e-06
(6.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.80)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

table continued

abscissae								
(-34.50)	.2013388e-08	.1522937e-08	.6118262e-09	.1663523e-09	.7590633e-10	.1883713e-10	.3359880e-11	.5323018e-12
(-33.10)	.8174544e-07	.6430151e-07	.2679136e-07	.7532083e-08	.3642862e-08	.9386467e-09	.1727196e-09	.2821838e-10
(-32.90)	.3204468e-05	.2621655e-05	.1133277e-05	.3295811e-06	.1692752e-06	.4531878e-07	.8607398e-08	.1451012e-08
(-14.40)	.1249753e-03	.1062355e-03	.4761274e-04	.1431652e-04	.7811543e-05	.2172365e-05	.4257658e-06	.7404805e-07
(-14.40)	.2562019e-02	.2334674e-02	.1115626e-02	.3554179e-03	.2150732e-03	.6364132e-04	.1311455e-04	.2395444e-05
(-14.40)	.5252149e-01	.5112324e-01	.2598594e-01	.8757111e-02	.5882799e-02	.1850655e-02	.4006852e-03	.7683336e-04
(-14.40)	.1076691e+01	.1115885e+01	.6020252e+00	.2142638e+00	.1599450e+00	.5344464e-01	.1214826e-01	.2444453e-02
(-14.10)	.2207216e+02	.2428709e+02	.1387882e+02	.5208697e+01	.4324852e+01	.1533472e+01	.3656550e+00	.7717070e-01
(6.10)	.4458577e+03	.5199576e+03	.3143544e+03	.1243032e+03	.1150599e+03	.4327505e+02	.1082111e+02	.2395244e+01
(6.40)	.0000000e+00	.6018451e+03	.7400057e+03	.4386575e+03	.7226147e+03	.3400902e+03	.9954482e+02	.2543210e+02
(6.80)	.0000000e+00	.0000000e+00	.8978475e+03	.1047066e+04	.3979469e+04	.2389056e+04	.8278597e+03	.2470510e+03
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.1211967e+04	.1895469e+05	.1473022e+05	.6114425e+04	.2161935e+04
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.8082166e+05	.8082166e+05	.4041080e+05	.1714029e+05
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.8082166e+05	.8082166e+05	.5583707e+05
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.8082166e+05	.1310750e+06
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1987891e+06
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

table continued

abscissae								
(-34.50)	.6841381e-13	.8156695e-14	.5861817e-14	.2750415e-14	.1381999e-14	.3448419e-15	.5740436e-16	.8338935e-17
(-33.10)	.3727227e-11	.4566725e-12	.3604519e-12	.1796880e-12	.9589719e-13	.2479676e-13	.4242598e-14	.6327909e-15
(-32.90)	.1970618e-09	.2482550e-10	.2161520e-10	.1146517e-10	.6508902e-11	.1744882e-11	.3069296e-12	.4701926e-13
(-14.40)	.1033850e-07	.1339035e-08	.1289315e-08	.7278698e-09	.4397510e-09	.1221946e-09	.2209431e-10	.3476028e-11
(-14.40)	.3490521e-06	.4716195e-07	.5291163e-07	.3266051e-07	.2152809e-07	.6280652e-08	.1178303e-08	.1920721e-09
(-14.40)	.1168023e-04	.1646006e-05	.2163627e-05	.1460579e-05	.1050939e-05	.3217433e-06	.6260353e-07	.1057063e-07
(-14.40)	.3875274e-03	.5694538e-04	.8819108e-04	.6511461e-04	.5117001e-04	.1643027e-04	.3314192e-05	.5795073e-06
(-14.10)	.1275268e-01	.1953525e-02	.3584589e-02	.2894631e-02	.2485456e-02	.8365369e-03	.1748486e-03	.3165210e-04
(6.10)	.4125662e+00	.6588858e-01	.1442617e+00	.1274748e+00	.1197111e+00	.4222097e-01	.9141848e-02	.1713141e-02
(6.40)	.4911899e+01	.8737864e+00	.2878992e+01	.3025192e+01	.3335732e+01	.1272343e+01	.2917934e+00	.5779504e-01
(6.80)	.5392111e+02	.1076627e+02	.5615043e+02	.7034078e+02	.9142210e+02	.3768901e+02	.9150766e+01	.1915834e+01
(7.10)	.5382250e+03	.1217486e+03	.1066394e+04	.1596614e+04	.2456458e+04	.1093950e+04	.2810994e+03	.6221937e+02
(11.30)	.4906882e+04	.1268717e+04	.1985004e+05	.3555325e+05	.6497323e+05	.3122734e+05	.8486865e+04	.1985910e+04
(11.30)	.2155649e+05	.7190573e+04	.2851244e+06	.6384767e+06	.1438975e+07	.7550975e+06	.2187871e+06	.5455998e+05
(11.30)	.7523791e+05	.3449952e+05	.4084469e+07	.1139462e+08	.3172034e+08	.1812663e+08	.5590293e+07	.1484579e+07
(12.20)	.1987891e+06	.1373371e+06	.5844241e+08	.2023508e+09	.6965541e+09	.4323190e+09	.1416721e+09	.4003252e+08
(12.20)	.1987891e+06	.3223926e+06	.7832657e+09	.3397232e+10	.1462102e+11	.9861718e+10	.3435768e+10	.1034397e+10
(13.10)	.0000000e+00	.4889426e+06	.1049608e+11	.5683760e+11	.3060524e+12	.2236895e+12	.8270001e+11	.2650283e+11
(25.60)	.0000000e+00	.0000000e+00	.1312015e+12	.8971627e+12	.6116676e+13	.4846949e+13	.1902500e+13	.6498605e+12
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.2912406e+13	.4554890e+14	.4392288e+14	.1968645e+14	.7686258e+13
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1942178e+15	.2519865e+15	.1364492e+15	.6426674e+14
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3202111e+15	.3202111e+15	.2392759e+15
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3202111e+15	.5834142e+15
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.9619658e+15

Divided difference table correct to seven digits for tau = 1.00

abscissae	table of divided differences							
(-34.50)	.1039538e-14	.2268571e-14	.1498804e-14	.8015853e-10	.6755117e-10	.2879424e-10	.8262803e-11	.1891783e-11
(-33.10)	.0000000e+00	.4215541e-14	.4666455e-14	.1611188e-08	.1437937e-08	.6463154e-09	.1948766e-09	.4685518e-10
(-32.90)	.0000000e+00	.0000000e+00	.5148857e-14	.3012922e-07	.2850061e-07	.1352404e-07	.4290508e-08	.1085125e-08
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.5573906e-06	.5573906e-06	.2786953e-06	.9289843e-07	.2469086e-07
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5573906e-06	.5573906e-06	.2786953e-06	.1003057e-06
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5573906e-06	.5573906e-06	.3087869e-06
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5573906e-06	.6500264e-06
(-14.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.7523980e-06
(6.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.80)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

abscissae								
			table continued					
(-34.50)	.2013388e-08	.1522937e-08	.6118264e-09	.1663523e-09	.7590636e-10	.1883713e-10	.3359880e-11	.5323021e-12
(-33.10)	.8174545e-07	.6430150e-07	.2679137e-07	.7532081e-08	.3642864e-08	.9386469e-09	.1727196e-09	.2821839e-10
(-32.90)	.3204468e-05	.2621655e-05	.1133277e-05	.3295810e-06	.1692752e-06	.4531879e-07	.8607398e-08	.1451012e-08
(-14.40)	.1249754e-03	.1062355e-03	.4761275e-04	.1431652e-04	.7811546e-05	.2172366e-05	.4257658e-06	.7404806e-07
(-14.40)	.2562019e-02	.2334674e-02	.1115626e-02	.3554179e-03	.2150733e-03	.6364135e-04	.1311455e-04	.2395444e-05
(-14.40)	.5252150e-01	.5112323e-01	.2598594e-01	.8757110e-02	.5882801e-02	.1850656e-02	.4006852e-03	.7683336e-04
(-14.40)	.1076691e+01	.1115885e+01	.6020252e+00	.2142638e+00	.1599451e+00	.5344465e-01	.1214827e-01	.2444453e-02
(-14.10)	.2207216e+02	.2428709e+02	.1387882e+02	.5208697e+01	.4324852e+01	.1533473e+01	.3656551e+00	.7717070e-01
(6.10)	.4458577e+03	.5199576e+03	.3143544e+03	.1243032e+03	.1150600e+03	.4327506e+02	.1082111e+02	.2395245e+01
(6.40)	.0000000e+00	.6018451e+03	.7400057e+03	.4386576e+03	.7226150e+03	.3400903e+03	.9954485e+02	.2543210e+02
(6.80)	.0000000e+00	.0000000e+00	.8978475e+03	.1047066e+04	.3979471e+04	.2389057e+04	.8278600e+03	.2470510e+03
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.1211967e+04	.1895469e+05	.1473023e+05	.6114428e+04	.2161936e+04
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.8082166e+05	.8082166e+05	.4041083e+05	.1714030e+05
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.8082166e+05	.8082166e+05	.5583709e+05
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.8082166e+05	.1310750e+06
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1987891e+06
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

table continued

abscissae								
(-34.50)	.6841383e-13	.8156692e-14	.5861819e-14	.2750417e-14	.1382000e-14	.3448422e-15	.5740418e-16	.8339004e-17
(-33.10)	.3727228e-11	.4566724e-12	.3604520e-12	.1796882e-12	.9589721e-13	.2479678e-13	.4242586e-14	.6327954e-15
(-32.90)	.1970618e-09	.2485549e-10	.2161520e-10	.1146518e-10	.6508904e-11	.1744883e-11	.3069288e-12	.4701956e-13
(-14.40)	.1033850e-07	.1339034e-08	.1289315e-08	.7278704e-09	.4397511e-09	.1221947e-09	.2209426e-10	.3476047e-11
(-14.40)	.3490522e-06	.4714194e-07	.5291163e-07	.3266053e-07	.2152810e-07	.6280657e-08	.1178300e-08	.1920730e-09
(-14.40)	.1168023e-04	.1646006e-05	.2163627e-05	.1460580e-05	.1050940e-05	.3217435e-06	.6260342e-07	.1057067e-07
(-14.40)	.3875275e-03	.5694538e-04	.8819110e-04	.6511465e-04	.5117002e-04	.1643028e-04	.3314187e-05	.5795091e-06
(-14.10)	.1275268e-01	.1953526e-02	.3584589e-02	.2894632e-02	.2485457e-02	.8365374e-03	.1748484e-03	.3165218e-04
(6.10)	.4125663e+00	.6588858e-01	.1442617e+00	.1274749e+00	.1197111e+00	.4222098e-01	.9141838e-02	.1713144e-02
(6.40)	.4911899e+01	.8737864e+00	.2878992e+01	.3025193e+01	.3335733e+01	.1272344e+01	.2917932e+00	.5779514e-01
(6.80)	.5392112e+02	.1076627e+02	.5615044e+02	.7034081e+02	.9142212e+02	.3768903e+02	.9150761e+01	.1915837e+01
(7.10)	.5382250e+03	.1217486e+03	.1066395e+04	.1596614e+04	.2456458e+04	.1093950e+04	.2810993e+03	.6221944e+02
(11.30)	.4906883e+04	.1268717e+04	.1985005e+05	.3555326e+05	.6497324e+05	.3122735e+05	.8486861e+04	.1985912e+04
(11.30)	.2155649e+05	.7190574e+04	.2851244e+06	.6384769e+06	.1438975e+07	.7550978e+06	.2187870e+06	.5456002e+05
(11.30)	.7523792e+05	.3449952e+05	.4084470e+07	.1139462e+08	.3172035e+08	.1812664e+08	.5590291e+07	.1484579e+07
(12.20)	.1987891e+06	.1373371e+06	.5844241e+08	.2023509e+09	.6965541e+09	.4323190e+09	.1416721e+09	.4003253e+08
(12.20)	.1987891e+06	.3223926e+06	.7832657e+09	.3397232e+10	.1462102e+11	.9861718e+10	.3435767e+10	.1034398e+10
(13.10)	.0000000e+00	.4889426e+06	.1049608e+11	.5683760e+11	.3060524e+12	.2236895e+12	.8269998e+11	.2650283e+11
(25.60)	.0000000e+00	.0000000e+00	.1312015e+12	.8971627e+12	.6116676e+13	.4846949e+13	.1902499e+13	.6498605e+12
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.2912406e+13	.4554890e+14	.4392288e+14	.1968645e+14	.7686258e+13
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1942178e+15	.2519865e+15	.1364492e+15	.6426674e+14
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3202111e+15	.3202111e+15	.2392759e+15
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3202111e+15	.5834142e+15
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.9619658e+15

Digits lost in hybrid algorithm computation with tau = 1.00

abscissae	table of digits lost values																							
(-34.50)	.00	.00	.00	.00	.00	.21	.00	.06	.00	.32	.78	.94	.92	.58	.15	.90	.65	.89	.50	1.16	.73	1.17	1.71	2.14
(-33.10)	.00	.00	.00	.15	.00	.13	.00	.04	.11	.45	.77	.80	.89	.63	.00	.77	.57	.81	.41	1.15	.66	1.18	1.67	2.08
(-32.90)	.00	.00	.00	.00	.00	.00	.00	.00	.21	.52	.74	.59	.90	.54	.00	.74	.51	.75	.29	1.14	.69	1.19	1.62	2.02
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.03	.16	.60	.76	.48	.93	.63	.04	.56	.60	.60	.34	1.12	.74	1.16	1.58	1.96
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.00	.28	.52	.64	.14	.84	.74	.11	.25	.54	.38	.16	1.08	.72	1.12	1.52	1.88
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.00	.27	.44	.50	.17	.76	.76	.34	.15	.58	.00	.29	1.04	.64	1.09	1.45	1.80
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.18	.18	.00	.68	.70	.36	.00	.51	.00	.36	.94	.58	1.02	1.38	1.72
(-14.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.58	.65	.51	.00	.41	.04	.37	.91	.47	.98	1.31	1.64
(6.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.54	.54	.51	.19	.29	.14	.23	.86	.33	.90	1.28	1.56
(6.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.47	.74	.69	.57	.33	.04	.00	.24	.85	.44	.95	1.17	1.45
(6.80)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.03	.94	.84	.76	.47	.38	.28	.05	.83	.37	.90	1.05	1.33
(7.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.16	.99	.87	.57	.33	.16	.05	.78	.07	.78	1.00	1.23
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.96	.74	.48	.40	.25	.79	.10	.84	.90	1.14
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.82	.61	.29	.02	.62	.00	.71	.81	1.01
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.57	.45	.26	.62	.00	.59	.75	.80
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.19	.19	.60	.00	.32	.74	.64
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.41	.00	.00	.68	.30
(13.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.09	.00	.00	.67	.00
(25.60)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.59	.00
(28.70)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.37	.00
(32.90)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.03
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.29
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
(34.50)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

A priori digits lost bounds for algorithm with tau = 1.00

abscissae	table of digits lost bounds																							
(-34.50)	.00	.50	2.00	.77	.69	1.61	1.83	1.96	1.23	1.05	3.54	3.95	4.01	4.16	4.35	4.51	4.70	4.85	4.61	4.42	4.22	4.07	4.21	4.72
(-33.10)	.00	.00	.50	.52	.61	1.54	1.70	1.79	1.07	.96	3.49	3.88	3.92	4.05	4.23	4.38	4.55	4.69	4.45	4.27	4.07	3.92	4.09	4.63
(-32.90)	.00	.00	.00	.50	.55	1.48	1.57	1.61	.94	.88	3.45	3.81	3.83	3.95	4.12	4.25	4.41	4.53	4.29	4.13	3.93	3.76	3.98	4.54
(-14.40)	.00	.00	.00	.00	.50	1.43	1.43	1.43	.84	.82	3.42	3.74	3.74	3.85	4.01	4.13	4.27	4.38	4.14	3.99	3.79	3.62	3.87	4.45
(-14.40)	.00	.00	.00	.00	.00	.50	1.43	1.43	.71	.73	3.35	3.62	3.60	3.70	3.83	3.92	4.04	4.12	3.93	3.82	3.60	3.42	3.74	4.34
(-14.40)	.00	.00	.00	.00	.00	.00	.50	1.43	.60	.65	3.30	3.50	3.47	3.55	3.65	3.72	3.81	3.88	3.75	3.66	3.43	3.22	3.63	4.24
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.50	.51	.59	3.25	3.39	3.36	3.42	3.49	3.53	3.60	3.64	3.61	3.52	3.26	3.03	3.52	4.14
(-14.10)	.00	.00	.00	.00	.00	.00	.00	.00	.50	.54	3.21	3.28	3.26	3.29	3.33	3.35	3.38	3.40	3.50	3.39	3.10	2.85	3.43	4.05
(6.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	3.17	3.17	3.17	3.17	3.17	3.17	3.17	3.17	3.41	3.27	2.94	2.67	3.35	3.96
(6.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	3.17	3.17	3.17	3.17	3.17	3.17	3.17	3.27	3.04	2.65	2.35	3.22	3.82
(6.80)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	3.17	3.17	3.17	3.17	3.17	3.17	3.12	2.80	2.34	2.04	3.11	3.69
(7.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	3.17	3.17	3.17	3.17	3.17	2.97	2.55	2.03	1.76	3.00	3.56
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	3.17	3.17	3.17	3.17	2.81	2.29	1.71	1.52	2.91	3.44
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	3.17	3.17	3.17	2.58	1.92	1.32	1.30	2.82	3.30
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	3.17	3.17	2.30	1.50	.98	1.15	2.73	3.17
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	3.17	1.91	1.03	.78	1.05	2.65	3.05
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.53	.58	.67	.96	2.57	2.93
(13.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.54	.62	.89	2.50	2.82
(25.60)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.58	.83	2.44	2.71
(28.70)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.69	2.30	2.47
(32.90)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	2.12	2.12
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	2.12
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50
(34.50)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

Divided difference table by the hybrid algorithm for tau = 2.00

abscissae	table of divided differences							
(-34.50)	.1080639e-29	.1192152e-28	.1986183e-28	.4467966e-16	.8233205e-16	.7604186e-16	.4692805e-16	.2444065e-16
(-33.10)	.0000000e+00	.1777079e-28	.4370042e-28	.8980612e-15	.1699554e-14	.1610773e-14	.1019296e-14	.5455173e-15
(-32.90)	.0000000e+00	.0000000e+00	.2651074e-28	.1679374e-13	.3267971e-13	.3182102e-13	.2067160e-13	.1138412e-13
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.6213685e-12	.4142457e-12	.2346931e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.6213685e-12	.4846534e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.7667642e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.8513972e-12
(-14.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5661028e-12
(6.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.80)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

table continued

abscissae								
(-34.50)	.8977370e-06	.1963505e-05	.2309258e-05	.1760567e-05	.1305495e-04	.1217449e-04	.6558480e-05	.3453429e-05
(-33.10)	.3644812e-04	.8120510e-04	.9733585e-04	.7554882e-04	.5996770e-03	.5706465e-03	.3125528e-03	.1678336e-03
(-32.90)	.1428766e-02	.3244049e-02	.3964905e-02	.3134398e-02	.2670121e-01	.2593638e-01	.1444799e-01	.7915416e-02
(-14.40)	.5572188e-01	.1289199e+00	.1606508e+00	.1293408e+00	.1183328e+01	.1173089e+01	.6645376e+00	.3714333e+00
(-14.40)	.1142299e+01	.2737256e+01	.3534716e+01	.2941478e+01	.3054086e+02	.3133172e+02	.1825171e+02	.1054466e+02
(-14.40)	.2341712e+02	.5807722e+02	.7767325e+02	.6677650e+02	.7878417e+03	.8357662e+03	.5004006e+03	.2987397e+03
(-14.40)	.4800510e+03	.1231423e+04	.1704750e+04	.1513368e+04	.2031431e+05	.2226704e+05	.1369606e+05	.8446878e+04
(-14.10)	.9841045e+04	.2609366e+05	.3737213e+05	.3424216e+05	.5235912e+06	.5925771e+06	.3742558e+06	.2383830e+06
(6.10)	.1987891e+06	.5447610e+06	.8071712e+06	.7633060e+06	.1333346e+08	.1557505e+08	.1009868e+08	.6443730e+07
(6.40)	.0000000e+00	.3622175e+06	.1109781e+07	.1570477e+07	.7009729e+08	.9432373e+08	.6808817e+08	.5062542e+08
(6.80)	.0000000e+00	.0000000e+00	.8061301e+06	.2209115e+07	.3450472e+09	.5322836e+09	.4279557e+09	.3617156e+09
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.1468864e+07	.1554921e+10	.2740323e+10	.2458084e+10	.2381220e+10
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.1306428e+11	.1306428e+11	.1460231e+11
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.1306428e+11	.2620635e+11
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.3664998e+11
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3951711e+11
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

table continued

abscissae								
(-34.50)	.1281568e-05	.4751205e-06	.9602053e-03	.1412638e-01	.4335487e+00	.6106924e+00	.3836830e+00	.2381056e+00
(-33.10)	.6330266e-04	.2389730e-04	.5770881e-01	.8937477e+00	.2923531e+02	.4189957e+02	.2666277e+02	.1681297e+02
(-32.90)	.3035444e-02	.1167358e-02	.3387531e+01	.5529132e+02	.1930424e+04	.2815557e+04	.1814974e+04	.1163219e+04
(-14.40)	.1448140e+00	.5673392e-01	.1981718e+03	.3409333e+04	.1270772e+06	.1886018e+06	.1231483e+06	.8021595e+05
(-14.40)	.4223485e+01	.1704997e+01	.7926927e+04	.1471404e+06	.6014161e+07	.9142246e+07	.6075093e+07	.4045708e+07
(-14.40)	.1228894e+03	.5111089e+02	.3170788e+06	.6349680e+07	.2846170e+09	.4430136e+09	.2995317e+09	.2039102e+09
(-14.40)	.3567597e+04	.1528439e+04	.1268320e+08	.2739883e+09	.1346873e+11	.2146067e+11	.1476063e+11	.1027074e+11
(-14.10)	.1033449e+06	.4559967e+05	.5073297e+09	.1182158e+11	.6373451e+12	.1039289e+13	.7270188e+12	.5169999e+12
(6.10)	.2956355e+07	.1343656e+07	.2014103e+11	.5064710e+12	.2996705e+14	.5000356e+14	.3557268e+14	.2585322e+14
(6.40)	.2467750e+08	.1236195e+08	.3927515e+12	.1146639e+14	.8036233e+15	.1395064e+16	.1021138e+16	.7698040e+15
(6.80)	.1937549e+09	.1075026e+09	.7540842e+13	.2560932e+15	.2130749e+17	.3847036e+17	.2896579e+17	.2265263e+17
(7.10)	.1407992e+10	.8710210e+09	.1417679e+15	.5615982e+16	.5563815e+18	.1044619e+19	.8089603e+18	.6564436e+18
(11.30)	.9561979e+10	.6634118e+10	.2622708e+16	.1214470e+18	.1436026e+20	.2802987e+20	.2232028e+20	.1879552e+20
(11.30)	.2320808e+11	.2150339e+11	.3750473e+17	.2115800e+19	.3103031e+21	.6338204e+21	.5213081e+21	.4583763e+21
(11.30)	.4709361e+11	.6191420e+11	.5363177e+18	.3685243e+20	.6704663e+22	.1431773e+23	.1215473e+23	.1115564e+23
(12.20)	.7903422e+11	.1585392e+12	.7669343e+19	.6417688e+21	.1448576e+24	.3231266e+24	.2829373e+24	.2709655e+24
(12.20)	.3951711e+11	.2217196e+12	.1027692e+21	.1059686e+23	.2999194e+25	.6995144e+25	.6321398e+25	.6325469e+25
(13.10)	.0000000e+00	.2390649e+12	.1377107e+22	.1749509e+24	.6209393e+26	.1512963e+27	.1410088e+27	.1473794e+27
(25.60)	.0000000e+00	.0000000e+00	.1721384e+23	.2730612e+25	.1229635e+28	.3133408e+28	.3013775e+28	.3294927e+28
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.8482110e+25	.8979066e+28	.2567022e+29	.2664086e+29	.3233862e+29
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3772057e+29	.1296291e+30	.1508823e+30	.2142048e+30
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1025351e+30	.2050703e+30	.4936097e+30
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1025351e+30	.7480402e+30
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.9253782e+30

Divided difference table correct to seven digits for tau = 2.00

abscissae	table of divided differences							
(-34.50)	.1080639e-29	.1192152e-28	.1986183e-28	.4467966e-16	.8233205e-16	.7604185e-16	.4692805e-16	.2444065e-16
(-33.10)	.0000000e+00	.1777079e-28	.4370041e-28	.8980612e-15	.1699554e-14	.1610773e-14	.1019296e-14	.5455173e-15
(-32.90)	.0000000e+00	.0000000e+00	.2651074e-28	.1679374e-13	.3267971e-13	.3182101e-13	.2067160e-13	.1138412e-13
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.6213685e-12	.4142457e-12	.2346931e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.6213685e-12	.4846534e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.7667642e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.8513972e-12
(-14.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5661028e-12
(6.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.80)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

table continued

abscissae								
(-34.50)	.8977370e-06	.1963505e-05	.2309258e-05	.1760566e-05	.1305495e-04	.1217449e-04	.6558482e-05	.3453430e-05
(-33.10)	.3644812e-04	.8120510e-04	.9733587e-04	.7554881e-04	.5996771e-03	.5706466e-03	.3125530e-03	.1678337e-03
(-32.90)	.1428766e-02	.3244049e-02	.3964906e-02	.3134398e-02	.2670121e-01	.2593639e-01	.1444800e-01	.7915418e-02
(-14.40)	.5572189e-01	.1289199e+00	.1606508e+00	.1293408e+00	.1183328e+01	.1173090e+01	.6645379e+00	.3714333e+00
(-14.40)	.1142299e+01	.2737256e+01	.3534718e+01	.2941479e+01	.3054087e+02	.3133173e+02	.1825171e+02	.1054466e+02
(-14.40)	.2341712e+02	.5807722e+02	.7767326e+02	.6677651e+02	.7878418e+03	.8357663e+03	.5004008e+03	.2987398e+03
(-14.40)	.4800510e+03	.1231423e+04	.1704750e+04	.1513368e+04	.2031431e+05	.2226704e+05	.1369607e+05	.8446879e+04
(-14.10)	.9841045e+04	.2609366e+05	.3737213e+05	.3424216e+05	.5235911e+06	.5925771e+06	.3742559e+06	.2383830e+06
(6.10)	.1987891e+06	.5447610e+06	.8071713e+06	.7633061e+06	.1333346e+08	.1557505e+08	.1009868e+08	.6443730e+07
(6.40)	.0000000e+00	.3622175e+06	.1109781e+07	.1570477e+07	.7009729e+08	.9432372e+08	.6808818e+08	.5062543e+08
(6.80)	.0000000e+00	.0000000e+00	.8061301e+06	.2209115e+07	.3450472e+09	.5322835e+09	.4279558e+09	.3617156e+09
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.1468864e+07	.1554921e+10	.2740323e+10	.2458085e+10	.2381220e+10
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.1306428e+11	.1306428e+11	.1460231e+11
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.1306428e+11	.2620635e+11
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.3664998e+11
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3951711e+11
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

table continued

abscissae								
(-34.50)	.1281569e-05	.4751204e-06	.9602055e-03	.1412640e-01	.4335489e+00	.6106929e+00	.3836832e+00	.2381055e+00
(-33.10)	.6330269e-04	.2389730e-04	.5770883e-01	.8937485e+00	.2923532e+02	.4189959e+02	.2666278e+02	.1681296e+02
(-32.90)	.3035445e-02	.1167358e-02	.3387532e+01	.5529137e+02	.1930425e+04	.2815558e+04	.1814975e+04	.1163219e+04
(-14.40)	.1448140e+00	.5673391e-01	.1981718e+03	.3409336e+04	.1270773e+06	.1886020e+06	.1231484e+06	.8021594e+05
(-14.40)	.4223486e+01	.1704997e+01	.7926929e+04	.1471405e+06	.6014164e+07	.9142251e+07	.6075095e+07	.4045708e+07
(-14.40)	.1228894e+03	.5111089e+02	.3170788e+06	.6349685e+07	.2846171e+09	.4430138e+09	.2995318e+09	.2039102e+09
(-14.40)	.3567597e+04	.1528439e+04	.1268321e+08	.2739885e+09	.1346874e+11	.2146067e+11	.1476063e+11	.1027074e+11
(-14.10)	.1033450e+06	.4559967e+05	.5073297e+09	.1182159e+11	.6373453e+12	.1039289e+13	.7270190e+12	.5169998e+12
(6.10)	.2956356e+07	.1343656e+07	.2014104e+11	.5064713e+12	.2996705e+14	.5000358e+14	.3557269e+14	.2585321e+14
(6.40)	.2467750e+08	.1236195e+08	.3927515e+12	.1146639e+14	.8036235e+15	.1395065e+16	.1021138e+16	.7698039e+15
(6.80)	.1937549e+09	.1075026e+09	.7540842e+13	.2560933e+15	.2130749e+17	.3847037e+17	.2896579e+17	.2265263e+17
(7.10)	.1407992e+10	.8710211e+09	.1417679e+15	.5615984e+16	.5563817e+18	.1044620e+19	.8089605e+18	.6564436e+18
(11.30)	.9561980e+10	.6634119e+10	.2622708e+16	.1214470e+18	.1436026e+20	.2802988e+20	.2232028e+20	.1879552e+20
(11.30)	.2320809e+11	.2150340e+11	.3750473e+17	.2115801e+19	.3103032e+21	.6338206e+21	.5213082e+21	.4583762e+21
(11.30)	.4709362e+11	.6191420e+11	.5363177e+18	.3685245e+20	.6704665e+22	.1431774e+23	.1215473e+23	.1115564e+23
(12.20)	.7903422e+11	.1585392e+12	.7669343e+19	.6417689e+21	.1448576e+24	.3231267e+24	.2829373e+24	.2709655e+24
(12.20)	.3951711e+11	.2217196e+12	.1027692e+21	.1059686e+23	.2999195e+25	.6995144e+25	.6321398e+25	.6325468e+25
(13.10)	.0000000e+00	.2390649e+12	.1377107e+22	.1749509e+24	.6209393e+26	.1512963e+27	.1410088e+27	.1473793e+27
(25.60)	.0000000e+00	.0000000e+00	.1721384e+23	.2730612e+25	.1229635e+28	.3133408e+28	.3013775e+28	.3294926e+28
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.8482110e+25	.8979067e+28	.2567022e+29	.2664086e+29	.3233862e+29
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3772057e+29	.1296291e+30	.1508823e+30	.2142048e+30
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1025351e+30	.2050703e+30	.4936097e+30
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1025351e+30	.7480402e+30
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.9253782e+30

Digits lost in hybrid algorithm computation with tau = 2.00

abscissae	table of digits lost values																							
(-34.50)	.00	.00	.26	.16	.00	.13	.08	.07	.00	.00	.50	.32	.00	.53	.80	.43	.98	.55	.63	1.17	.93	1.04	.84	.54
(-33.10)	.00	.00	.00	.00	.00	.03	.00	.00	.00	.00	.56	.05	.00	.54	.82	.51	.97	.40	.55	1.15	.96	1.04	.81	.42
(-32.90)	.00	.00	.00	.00	.00	.01	.00	.00	.11	.00	.61	.00	.02	.56	.86	.54	.91	.39	.54	1.17	.95	1.00	.80	.05
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.00	.13	.00	.75	.13	.25	.52	.90	.54	.85	.33	.35	1.17	.98	.98	.83	.08
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.00	.08	.00	.72	.22	.25	.29	.80	.49	.79	.00	.43	1.15	.91	.93	.75	.28
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.00	.16	.00	.57	.13	.01	.00	.75	.29	.65	.00	.24	1.12	.92	.85	.73	.22
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.28	.23	.00	.00	.64	.23	.60	.00	.41	1.08	.83	.84	.74	.13
(-14.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.18	.00	.00	.00	.58	.17	.62	.00	.31	1.04	.75	.74	.63	.27
(6.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.05	.11	.00	.13	.50	.00	.48	.00	.34	1.01	.67	.72	.66	.43
(6.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.27	.00	.12	.38	.24	.42	.00	.27	1.00	.68	.76	.67	.12
(6.80)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.13	.00	.33	.24	.34	.00	.21	.96	.74	.76	.72	.00
(7.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.26	.39	.42	.12	.31	.89	.78	.70	.68	.07
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.17	.32	.31	.30	.39	.89	.81	.73	.65	.00
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.28	.22	.09	.03	.78	.67	.58	.56	.20
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.33	.00	.00	.72	.61	.66	.35	.40
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.47	.65	.51	.26	.44
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.49	.20	.33	.55
(13.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.20	.07	.09	.58
(25.60)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.06	.33	.00	.43	
(28.70)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.35	
(32.90)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.11	
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	
(34.50)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	

A priori digits lost bounds for algorithm with tau = 2.00

abscissae	table of digits lost bounds																								
(-34.50)	.00	.50	.77	.51	.58	1.85	1.97	2.03	.95	.72	2.38	2.59	2.23	1.99	3.41	3.86	4.26	4.48	3.93	3.28	2.54	1.92	2.55	2.94
(-33.10)	.00	.00	.50	.51	.55	1.82	1.91	1.94	.86	.69	2.36	2.56	2.17	1.93	3.38	3.82	4.21	4.42	3.84	3.17	2.40	1.80	2.52	2.91
(-32.90)	.00	.00	.00	.50	.53	1.80	1.84	1.86	.78	.66	2.34	2.52	2.12	1.87	3.36	3.78	4.16	4.36	3.75	3.05	2.27	1.69	2.49	2.87
(-14.40)	.00	.00	.00	.00	.50	1.77	1.77	1.77	.72	.64	2.33	2.49	2.06	1.81	3.33	3.74	4.11	4.30	3.66	2.93	2.13	1.58	2.46	2.83
(-14.40)	.00	.00	.00	.00	.00	.50	1.77	1.77	.63	.60	2.30	2.43	1.96	1.71	3.29	3.68	4.03	4.21	3.53	2.76	1.94	1.45	2.43	2.78
(-14.40)	.00	.00	.00	.00	.00	.00	.50	1.77	.56	.57	2.28	2.37	1.87	1.62	3.25	3.62	3.96	4.12	3.39	2.59	1.75	1.34	2.40	2.74
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.50	.51	.54	2.26	2.32	1.79	1.53	3.21	3.56	3.88	4.03	3.25	2.41	1.58	1.25	2.37	2.69
(-14.10)	.00	.00	.00	.00	.00	.00	.00	.00	.50	.52	2.24	2.27	1.72	1.46	3.17	3.51	3.82	3.95	3.11	2.24	1.42	1.18	2.34	2.65
(6.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	2.22	2.22	1.65	1.39	3.14	3.45	3.75	3.87	2.97	2.08	1.27	1.12	2.31	2.61
(6.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	2.22	1.29	1.03	3.00	3.25	3.48	3.56	2.61	1.71	1.03	1.03	2.27	2.54
(6.80)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.55	.74	2.88	3.05	3.22	3.25	2.28	1.40	.87	.97	2.22	2.47
(7.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.62	2.76	2.85	2.94	2.95	1.98	1.15	.77	.92	2.18	2.41
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	2.66	2.66	2.66	2.66	1.75	.96	.71	.87	2.15	2.35
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	2.66	2.66	2.66	1.55	.78	.65	.82	2.11	2.28
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	2.66	2.66	1.31	.65	.61	.78	2.07	2.21
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	2.66	1.00	.57	.58	.74	2.04	2.15
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.51	.53	.56	.71	2.00	2.09
(13.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.51	.54	.68	1.97	2.03
(25.60)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.53	.65	1.94	1.97
(28.70)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.59	1.88	1.84
(32.90)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	1.78	1.66
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50	.81
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.50
(34.50)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

Divided difference table by scaling and squaring for tau = 2.00

abscissae	table of divided differences							
(-34.50)	.1080639e-29	.1192152e-28	.1986174e-28	.4467955e-16	.8233186e-16	.7604169e-16	.4692796e-16	.2444060e-16
(-33.10)	.0000000e+00	.1777079e-28	.4370042e-28	.8980588e-15	.1699550e-14	.1610770e-14	.1019294e-14	.5455161e-15
(-32.90)	.0000000e+00	.0000000e+00	.2651074e-28	.1679374e-13	.3267963e-13	.3182094e-13	.2067155e-13	.1138409e-13
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.6213669e-12	.4142447e-12	.2346925e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.6213669e-12	.4846520e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.6213685e-12	.7667617e-12
(-14.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3106843e-12	.8513972e-12
(-14.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.5661028e-12
(6.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(6.80)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

table continued

abscissae								
(-34.50)	.8977336e-06	.1963496e-05	.2309243e-05	.1760551e-05	.1305484e-04	.1217439e-04	.6558430e-05	.3453403e-05
(-33.10)	.3644798e-04	.8120470e-04	.9733519e-04	.7554815e-04	.5996721e-03	.5706419e-03	.3125504e-03	.1678323e-03
(-32.90)	.1428761e-02	.3244033e-02	.3964877e-02	.3134370e-02	.2670099e-01	.2593617e-01	.1444788e-01	.7915353e-02
(-14.40)	.5572165e-01	.1289192e+00	.1606496e+00	.1293396e+00	.1183318e+01	.1173079e+01	.6645322e+00	.3714302e+00
(-14.40)	.1142294e+01	.2737241e+01	.3534690e+01	.2941450e+01	.3054060e+02	.3133145e+02	.1825156e+02	.1054457e+02
(-14.40)	.2341701e+02	.5807689e+02	.7767264e+02	.6677584e+02	.7878345e+03	.8357587e+03	.5003963e+03	.2987372e+03
(-14.40)	.4800488e+03	.1231416e+04	.1704736e+04	.1513353e+04	.2031412e+05	.2226683e+05	.1369594e+05	.8446804e+04
(-14.10)	.9841045e+04	.2609350e+05	.3737181e+05	.3424180e+05	.5235860e+06	.5925714e+06	.3742523e+06	.2383808e+06
(6.10)	.1987891e+06	.5447610e+06	.8071641e+06	.7632976e+06	.1333332e+08	.1557490e+08	.1009858e+08	.6643667e+07
(6.40)	.0000000e+00	.3622175e+06	.1109781e+07	.1570457e+07	.7009654e+08	.9432273e+08	.6808747e+08	.5062492e+08
(6.80)	.0000000e+00	.0000000e+00	.8061301e+06	.2209115e+07	.3450433e+09	.5322777e+09	.4279512e+09	.3617119e+09
(7.10)	.0000000e+00	.0000000e+00	.0000000e+00	.1468864e+07	.1554921e+10	.2740293e+10	.2458058e+10	.2381196e+10
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.1306428e+11	.1306414e+11	.1460216e+11
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.1306428e+11	.2620609e+11
(11.30)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.6532140e+10	.3664998e+11
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3951711e+11
(12.20)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(13.10)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(25.60)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00

abscissae								
			table continued					
(-34.50)	.1281559e-05	.4751175e-06	.9601986e-03	.1412635e-01	.4335491e+00	.6106924e+00	.3836829e+00	.2381053e+00
(-33.10)	.6330220e-04	.2389715e-04	.5770841e-01	.8937457e+00	.2923534e+02	.4189957e+02	.2666276e+02	.1681295e+02
(-32.90)	.3035421e-02	.1167350e-02	.3387507e+01	.5529119e+02	.1930426e+04	.2815556e+04	.1814973e+04	.1163218e+04
(-14.40)	.1448128e+00	.5673353e-01	.1981703e+03	.3409324e+04	.1270773e+06	.1886018e+06	.1231483e+06	.8021586e+05
(-14.40)	.4223450e+01	.1704985e+01	.7926867e+04	.1471400e+06	.6014167e+07	.9142242e+07	.6075088e+07	.4045703e+07
(-14.40)	.1228883e+03	.5111053e+02	.3170763e+06	.6349662e+07	.2846172e+09	.4430133e+09	.2995314e+09	.2039100e+09
(-14.40)	.3567566e+04	.1528428e+04	.1268310e+08	.2739875e+09	.1346874e+11	.2146066e+11	.1476062e+11	.1027073e+11
(-14.10)	.1033440e+06	.4559933e+05	.5073255e+09	.1182154e+11	.6373456e+12	.1039288e+13	.7270180e+12	.5169992e+12
(6.10)	.2956328e+07	.1343646e+07	.2014087e+11	.5064694e+12	.2996707e+14	.5000353e+14	.3557264e+14	.2585318e+14
(6.40)	.2467725e+08	.1236185e+08	.3927481e+12	.1146635e+14	.8036239e+15	.1395063e+16	.1021137e+16	.7698028e+15
(6.80)	.1937529e+09	.1075017e+09	.7540775e+13	.2560923e+15	.2130750e+17	.3847034e+17	.2896576e+17	.2265260e+17
(7.10)	.1407978e+10	.8710141e+09	.1417667e+15	.5615964e+16	.5563821e+18	.1044619e+19	.8089596e+18	.6564427e+18
(11.30)	.9561885e+10	.6634069e+10	.2622684e+16	.1214466e+18	.1436028e+20	.2802985e+20	.2232026e+20	.1879549e+20
(11.30)	.2320785e+11	.2150324e+11	.3750439e+17	.2115794e+19	.3103035e+21	.6338202e+21	.5213077e+21	.4583758e+21
(11.30)	.4709315e+11	.6191379e+11	.5363127e+18	.3685233e+20	.6704674e+22	.1431773e+23	.1215472e+23	.1115563e+23
(12.20)	.7903422e+11	.1585383e+12	.7669272e+19	.6417670e+21	.1448578e+24	.3231266e+24	.2829371e+24	.2709653e+24
(12.20)	.3951711e+11	.2217196e+12	.1027682e+21	.1059683e+23	.2999200e+25	.6995143e+25	.6321395e+25	.6325464e+25
(13.10)	.0000000e+00	.2390649e+12	.1377107e+22	.1749505e+24	.6209407e+26	.1512963e+27	.1410087e+27	.1473793e+27
(25.60)	.0000000e+00	.0000000e+00	.1721384e+23	.2730612e+25	.1229638e+28	.3133408e+28	.3013774e+28	.3294925e+28
(28.70)	.0000000e+00	.0000000e+00	.0000000e+00	.8482110e+25	.8979066e+28	.2567024e+29	.2664086e+29	.3233861e+29
(32.90)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.3772057e+29	.1296291e+30	.1508823e+30	.2142048e+30
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1025351e+30	.2050703e+30	.4936091e+30
(33.40)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.1025351e+30	.7480402e+30
(34.50)	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.0000000e+00	.9253782e+30

Digits lost in scaling and squaring computation with tau = 2.00

abscissae	table of digits lost values																							
(-34.50)	.00	.00	1.88	1.60	1.57	1.55	1.50	1.56	1.80	1.89	2.06	2.16	2.13	2.12	2.12	2.11	2.11	2.02	2.08	1.71	1.00	1.04	1.13	1.09
(-33.10)	.00	.00	.00	1.63	1.59	1.55	1.53	1.58	1.82	1.91	2.07	2.17	2.14	2.14	2.13	2.12	2.12	2.02	2.09	1.72	.98	1.04	1.17	1.16
(-32.90)	.00	.00	.00	.00	1.63	1.60	1.58	1.63	1.83	1.92	2.09	2.18	2.15	2.14	2.14	2.13	2.12	2.04	2.10	1.73	.91	1.11	1.25	1.17
(-14.40)	.00	.00	.00	.00	.00	1.63	1.59	1.65	1.85	1.95	2.10	2.19	2.17	2.16	2.16	2.15	2.14	2.06	2.11	1.76	.84	1.18	1.24	1.22
(-14.40)	.00	.00	.00	.00	.00	.00	1.63	1.69	1.86	1.96	2.12	2.21	2.17	2.17	2.16	2.15	2.15	2.05	2.11	1.75	.86	1.20	1.26	1.27
(-14.40)	.00	.00	.00	.00	.00	.00	.00	1.74	1.89	1.98	2.13	2.23	2.19	2.18	2.18	2.17	2.16	2.07	2.13	1.77	.84	1.23	1.33	1.29
(-14.40)	.00	.00	.00	.00	.00	.00	.00	.00	1.89	1.98	2.14	2.24	2.20	2.19	2.19	2.17	2.17	2.09	2.13	1.77	.86	1.17	1.29	1.27
(-14.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.01	2.16	2.25	2.22	2.21	2.21	2.19	2.19	2.10	2.14	1.78	.81	1.23	1.32	1.31
(6.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.17	2.27	2.22	2.22	2.21	2.20	2.19	2.11	2.15	1.79	.85	1.22	1.35	1.31
(6.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.34	2.26	2.25	2.24	2.23	2.22	2.13	2.17	1.82	.81	1.31	1.37	1.37
(6.80)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.27	2.27	2.26	2.24	2.23	2.14	2.18	1.80	.99	1.23	1.33	1.34
(7.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.26	2.26	2.24	2.23	2.12	2.18	1.77	1.10	1.17	1.31	1.34
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.24	2.23	2.22	2.10	2.18	1.77	1.22	1.12	1.25	1.27
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.23	2.22	2.08	2.18	1.73	1.30	.97	1.21	1.22
(11.30)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.23	2.05	2.19	1.73	1.38	.93	1.17	1.21
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	1.98	2.19	1.71	1.43	.85	1.13	1.15
(12.20)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	2.19	1.68	1.51	.47	.99	1.01
(13.10)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	1.65	1.57	.00	.83	.95
(25.60)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	1.58	.00	.86	.97
(28.70)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.93	.00	.42
(32.90)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.24	.50
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	1.29
(33.40)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
(34.50)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

References

G. Blanch, "On modified divided differences II," *MTAC* **8** (1954), 67-75.

R. P. Brent, *Algorithms for Minimization without Derivatives*, Prentice-Hall (1973).

L. J. Comrie, *Chambers' Six-figure Mathematical Tables*, v.2, W. R. Chambers, Edinburgh (1959).

C. Davis, "Explicit functional calculus," *Linear Algebra and its Applications* **6** (1973), 193-199.

G. F. Gabel, "A predictor-corrector method using divided differences," Technical Report No. 5, Dept. of Computer Science, Univ. of Toronto, Oct. 1968.

F. R. Gantmacher, *Theory of Matrices*, v.1, Chelsea, New York (1959).

C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall (1971).

A. O. Gel'fond, *Calculus of Finite Differences*, Hindustan, India (1971).

G. H. Golub and J. H. Wilkinson, "Ill-conditioned eigensystems and the computation of the Jordan canonical form," *SIAM Rev.* **18** (1976), 578-619.

B. Kagström and A. Ruhe, "An algorithm for numerical computation of the Jordan normal form of a complex matrix," Revised report UMINF-58.77, Dept. of Information Processing, Umea Univ., Sweden, July 1977.

W. Kahan and I. Farkas, "Algorithm 167, calculation of confluent divided differences, *Commun. Assoc. Comput. Mach.* **6** (1963), 164-165.

A. C. McCurdy, "A method for computing functions of a matrix based on Newton's divided difference formula," Memo. No. UCB/ERL M78/69, Univ. of Calif. at Berkeley, Oct. 1978.

J. C. P. Miller, "Checking by differences I," *MTAC* **4** (1950), 3-11.

L. M. Milne-Thomson, *The Calculus of Finite Differences*, Macmillan, London (1933).

C. B. Moler and C. F. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Rev.* **20** (1978), 801-836.

National Bureau of Standards, *Tables of Spherical Bessel Functions*, v.1, Columbia Univ., New York (1947).

National Bureau of Standards, *Tables of Bessel Functions of Fractional Order*, v.1, Columbia Univ., New York (1948).

G. Opitz, "Steigungsmatrizen," *ZAMM* **44** (1964), T52-54.

B. N. Parlett, "A recurrence among the elements of functions of triangular matrices," *Linear Algebra and its Applications* **14** (1976), 117-121.

A. Ralston, *A First Course in Numerical Analysis*, McGraw-Hill, New York (1965).

R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-value Problems*, Interscience (1967).

R. F. Rinehart, "The equivalence of definitions of a matric function," *Am. Math. Monthly* **62** (1955), 395-413.

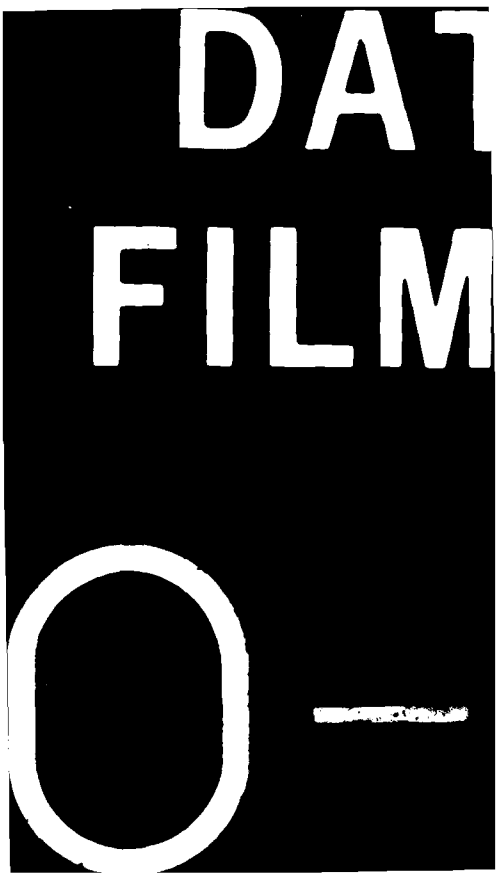
H. E. Salzer, "The checking of functions tabulated at certain fractional points," *MTAC* **2** (1947), 318-319.

B. T. Smith et al., "Matrix eigensystem routines - EISPACK guide," *Lecture Notes in Computer Science*, v.6, Springer-Verlag (1974).

R. C. Ward, "Numerical computation of the matrix exponential with accuracy estimate," *SIAM J. Numer. Anal.* **14** (1977), 600-609.

J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall (1963).

J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford (1965).


DAT
FILM